



Article

Survey and Classification of Automotive Security Attacks

Florian Sommer ^{*,†} , Jürgen Dürrwang ^{*,†}  and Reiner Kriesten ^{*}

Institute of Energy Efficient Mobility (IEEM), University of Applied Sciences, Moltkestrasse 30, 76133 Karlsruhe, Germany

* Correspondence: florian.sommer@hs-karlsruhe.de (F.S.); juergen.duerrwang@hs-karlsruhe.de (J.D.); reiner.kriesten@hs-karlsruhe.de (R.K.); Tel.: +49-721-925-1428 (F.S.); +49-721-925-1432 (J.D.); +49-721-925-1423 (R.K.)

† These authors contributed equally to this work.

Received: 1 April 2019; Accepted: 16 April 2019; Published: 19 April 2019



Abstract: Due to current development trends in the automotive industry towards stronger connected and autonomous driving, the attack surface of vehicles is growing which increases the risk of security attacks. This has been confirmed by several research projects in which vehicles were attacked in order to trigger various functions. In some cases these functions were critical to operational safety. To make automotive systems more secure, concepts must be developed that take existing attacks into account. Several taxonomies were proposed to analyze and classify security attacks. However, in this paper we show that the existing taxonomies were not designed for application in the automotive development process and therefore do not provide enough degree of detail for supporting development phases such as threat analysis or security testing. In order to be able to use the information that security attacks can provide for the development of security concepts and for testing automotive systems, we propose a comprehensive taxonomy with degrees of detail which addresses these tasks. In particular, our proposed taxonomy is designed in such a way, that each step in the vehicle development process can leverage it.

Keywords: automotive security; security attacks; classification; taxonomy

1. Introduction

The automotive industry is currently experiencing a major development trend towards connected and autonomous driving [1]. This has led to an increase in components installed in vehicles such as sensors, actuators, control units and communication systems. Especially, wireless interfaces such as wireless local area network (WLAN) [2] or Bluetooth [3], which are used for exchanging information between environment and vehicle, were implemented widely and changed vehicles from closed to open systems. This led to more complexity in vehicle communication and thus increased the attack surface for cyber attacks in which attackers can now access and influence vehicles from outside [4–13]. The numerous published attacks show not only that attack surfaces can be exploited, but also that attacks can influence the operational safety of a vehicle [14–17]. For this reason, vehicle security has reached original equipment manufacturers (OEMs) and suppliers.

As a result, security has been integrated into the vehicle development process [18,19]. One important part is the derivation of security requirements which usually result from a threat modelling carried out before. In threat modeling, a vehicle's attack surface is analyzed for attack possibilities. Attack paths are derived from the attack surface describing a path an attacker takes from the point of entry into the system through internal networks to target components. This information could also support the security testing process because security attacks can be used as test cases. Due to

that, attack paths can serve as detailed test steps. To accomplish these tasks, broad knowledge of automotive networks, safety and security is necessary. In order to support that process, a collection of possible attack paths would be helpful. It would also be valuable if complex incidents were decomposed into their individual steps in order to be able to reproduce all necessary steps of an attack. Unfortunately there is no established public knowledge database available for the automotive domain that contains existing incidents or possible attack scenarios. However, first approaches for automotive vulnerability databases have already been presented. The automotive information sharing and analysis center (AUTO-ISAC) [18] is an association of different vehicle manufacturers and suppliers. Within this collaboration information on vulnerabilities is collected and exchanged. However, this information is only accessible to specific members and therefore not public. Furthermore, upstream security [20] provides a public list of security attacks on vehicles, covering the period 2010 to 2019. Our colleagues from the Karlsruhe University of Applied Sciences [21] are working on developing of an automotive security vulnerability database as part of the security for connected, autonomous cars (SecForCARs) research project [22]. For developing such databases a large information base is necessary. As mentioned above, there are several publications of automotive security attacks as well as survey papers [9] which provide these information. The problem with these publications is that outlined attacks and scenarios are based on different approaches, executed in different test environments, and described in a different way.

As we show in Section 2, security taxonomies and vulnerability databases that describe security incidents and vulnerabilities exist mainly in the information technology (IT) area. However, there is the problem that these have not been designed for application in automotive development processes or security testing. Furthermore, they do not consider that vehicles are cyber-physical systems (CPSs) [23], based on control systems with sensors and actuators, that can have a significant impact on a vehicle's safety and its operating environment. As a result, they provide only a limited amount of information and are less suitable for supporting threat analysis and risk assessment (TARA) [18] or automotive security testing. Consequently, existent description models can usually not be transferred to the automotive domain. Thus, there is currently no publicly available description model for security attacks in the automotive sector.

Problem: existing security taxonomies are less suitable for an application in automotive development processes and therefore do not provide a uniform description of security incidents that can be understood by automotive engineers. As a result, existing attacks are not consistently described and classified in a way that allows the extraction of artefacts for vehicle development and security testing.

Approach: we provide a classification scheme to describe automotive security attacks as a uniform taxonomy. In order to be able to use attacks in several development stages as a source of information, we show how attacks can be represented in different levels of detail. For this purpose, we introduce hierarchical description levels for each category of our taxonomy.

Contribution: we present an attack taxonomy which is tailored to requirements of the automotive development process and allows attack composition and decomposition to identify possible attacks that did not occur in practice, yet. This includes:

- ★ A classification of 162 existing attacks based on our proposed taxonomy and an illustration of the value added by using it in the automotive security development process.
- ★ A decomposition of incidents that consist of multiple attack steps into individual blocks presented as a multi-stage attack path leading to a total of 413 attacks.
- ★ A rigorous discussion of the taxonomy and challenges that can arise when applying it in practice.

This work is structured as follows: in Section 2 we discuss existing automotive security attacks. Furthermore, we present known vulnerability databases and security taxonomies and discuss their transferability to the automotive sector. In Section 3 we present the classification categories of our approach. To illustrate how our taxonomy can be applied in practice, Section 4 contains a classification

of two exemplary attacks. In Section 5 we discuss our taxonomy critically and explain how we address possible problems. We also discuss quality criteria which should be fulfilled. Finally, in Section 5 we summarize our results and present planned future work.

2. Fundamentals and Related Work

This section provides a brief overview of automotive security vulnerabilities and existing vulnerability databases. In addition, known taxonomies for a description of security incidents, weaknesses and flaws are presented and their transferability to the automotive area is verified.

2.1. Automotive Security Attacks

Security attacks on vehicles have been shown through various publications we want to present here. Before the introduction of various network technologies such as Bluetooth or WLAN, vehicles could be regarded as closed systems. For this reason, security concepts focused on prevention of unauthorized access to vehicles (locking systems) and protection against theft (immobilizers). At that time, security attacks mainly consisted of overcoming locking systems and immobilizers. Such attacks became particularly relevant with the introduction of keyless entry systems [24]. The possibility of security attacks on such access systems has been demonstrated by various research papers [5,10–12,25–27]. The development of vehicles from closed to open systems has resulted in new attack surfaces an attacker can potentially exploit due to possible external communication. For this reason, researchers examined vehicles for the feasibility of such attacks [14,16,28,29]. Using local network access, researchers succeeded in compromising internal communication systems and influencing vehicle functions in an unauthorized manner. On the one hand, it was possible to manipulate functions such as window lifters, signal horns, indicator lights or displays [29]. On the other hand, researchers succeeded in triggering safety-critical functions like vehicle acceleration, brake or steering activation while driving, and lighting manipulations [14]. In the worst case, such attacks can lead to property damage and personal injury in road traffic. A further aspect at investigating security weak points is overcoming external communication interfaces. This allows manipulations from the outside via wireless connections, without having any local network access. In that area, there was a great contribution of research work [13,15,17,30–33] in which such attacks could be carried out. Due to the current development trend towards autonomous driving, the prevention of security attacks is of particular relevance, as drivers have no possibility to intervene in driving physics during highly autonomous driving [34]. For this reason, some researchers have devoted themselves to the investigation of security weaknesses of autonomous vehicles. Since these vehicles have to be equipped with a large number of sensors that detect the environment, Petit et al. [8] have been investigating such sensors. Remote attacks on camera and light imaging detection and ranging (LiDAR) systems were carried out. The researchers succeeded in blinding a camera and deceiving a LiDAR with fake echoes. Further work, for example by Sitawarin et al. [35] and Eykholt et al. [36], consisted of manipulating traffic sign recognition to simulate false traffic signs for self-driving vehicles. These examples of security attacks show a variety of possible attack scenarios in the automotive sector. However, the presented work is only a subset of existing attacks. For this reason, we collected automotive attack publications and applied our taxonomy to these attacks (attack list is available at [37]). In Section 5 we discuss that attack collection in more detail.

2.2. Vulnerability Databases

This section provides a brief overview of existing vulnerability databases. Managing security vulnerabilities is a major challenge and should be addressed as quickly as possible to mitigate attack chances. For this reason, vulnerabilities found or exploited in the past were published within vulnerability databases, especially in the IT sector. One of the best-known vulnerability databases is the national vulnerability database (NVD) [38] which currently contains 119,119 (as of March 2019) vulnerabilities covering the years 1988–2019. The NVD uses the common vulnerabilities and

exposures enumeration (CVE) [39] vulnerability list. A CVE entry contains an identifier, a description of a vulnerability and a reference to the vulnerability source. In addition to NVD, there are several other known vulnerability databases [40–45]. To our knowledge, these databases rarely address any automotive vulnerabilities. Only CVE describes individual known automotive security vulnerabilities [4,6]. That leads to the question whether existing databases are suitable for describing automotive vulnerabilities. The taxonomy presented in this work provides possible entries for a database.

2.3. Taxonomies

Taxonomies are used to classify objects according to certain criteria. As explained in [46], taxonomies depend on the area in which they are applied regarding to their structure and content. This can also be seen in the security field, where numerous taxonomies were developed and published in the past. These address different sub-areas, for example IT forensics [47]. We want to focus on security incident taxonomies in which vulnerability, attack and flaw classification can be distinguished [48]. As Weber et al. [48] mentioned in their work, different definitions for these terms exist in literature. Therefore, we agree on the definitions of Weber et al. Accordingly, a flaw denotes an error in a system which can lead to the violation of a security property. A vulnerability can be traced back to a flaw and describes a weakness in a system which can be successfully exploited by attacks. An attack (in this work the terms attack and incident are used as equal terms) represents a procedure for identifying or exploiting vulnerabilities. For the classification of flaws numerous works were published in the past [49–53]. Since flaws primarily arise during the development process, flaw taxonomies aim at supporting developers in finding and avoiding flaws. That includes developing analyzation tools like code analyzers [54] which examine software programs for potential errors. In this work, we are primarily concerned with automotive security incidents and their classification to support the development of security concepts as well as test activities. For this reason, attack and vulnerability taxonomies are presented below. A taxonomy for the classification of computer security intrusions was presented by Lindqvist and Jonsson [55]. This is based on the computer misuse techniques [56] presented by Neumann et al. which is divided into nine categories (N1–N9). The authors found that intrusions considered could only be classified in the categories N5 (bypass of intended controls), N6 (active misuse of resources) and N7 (passive misuse of resources). These were extended by subclasses in order to allow detailed descriptions of the intrusions. Since these categories refer to the view of system owners and administrators and only include three categories, that taxonomy is not suitable for our use cases.

In his dissertation [57] in 1997, John D. Howard presented a taxonomy to classify internet security incidents. There were 4299 security incidents, which occurred between 1989 and 1995, reported to the Computer Emergency Response Team/Coordination Center (CERT/CC) for investigation. Based on this work, Howard developed the Computer Emergency Response Team (CERT) taxonomy [58] in cooperation with Thomas A. Longstaff. That taxonomy finds a wide distribution in the IT sector. Seven categories are distinguished, each with different classification elements. In Figure 1 that classification model is shown.

The taxonomy in Figure 1 distinguishes three superior categories. The event describes the attack action and attack target. Above the event is the attack which includes event, tool, vulnerability and unauthorized result. Above the attack is the incident which includes the attack, attackers as well as the attacker's objectives. An application area of the CERT/CC taxonomy is IT forensics [47]. It is also used in the automotive sector. Hoppe et al. [59] presented a taxonomy that adapted the CERT model to the automotive sector. The superior categories event, attack and incident as well as the main categories Attackers, tool, vulnerability, action, target, unauthorized result and objectives were retained. Hoppe et al. added automotive relating sub-elements to individual categories. Furthermore, the CERT taxonomy has influence on other incident classification schemes from the automotive sector. The publication of Thing et al. [60] uses the categories attacker, attack vector, target,

motive and potential consequence to classify security incidents which can be seen as a subset of the CERT categories.

The taxonomies mentioned in this section describe attacks in an abstract way. Thus, the described incidents can be used for use cases like incident management. With a uniform description, attacks can be compared and represented abstractly without containing technical details. However, we want to address other use cases with our approach. In vehicle development, security attacks on vehicles are used to analyze a system for possible threats. These can be used to determine the risk of exploitability, derive security measures and perform security tests. For this reason, it would be useful to describe incidents in a way to address these processes. However, the existing taxonomies are too abstract for that, since they serve other use cases (incident management) and do not provide enough information for our purposes. With our taxonomy, we want to present a description of security attacks that is tailored to the automotive development process and covers the use cases TARA and security testing, so a more detailed description is necessary.

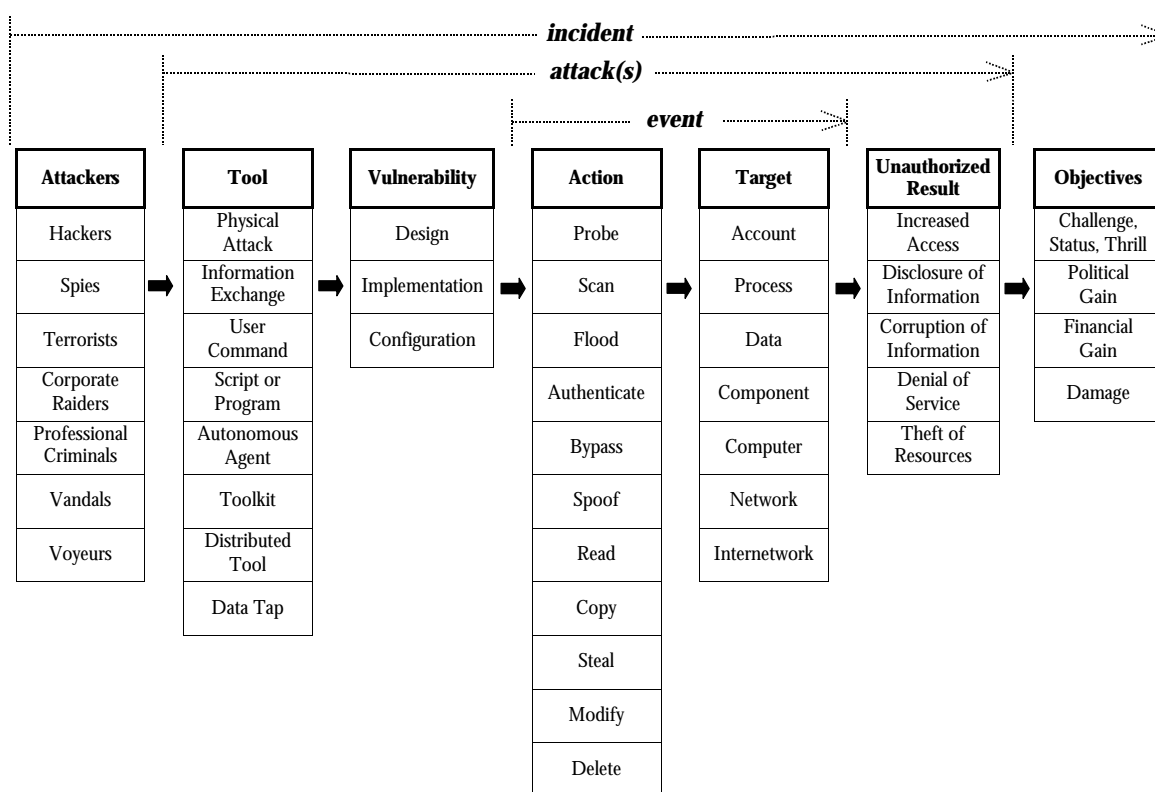


Figure 1. The Computer Emergency Response Team/Coordination Center (CERT/CC) taxonomy developed by Howard and Longstaff [58].

3. Automotive Security Taxonomy

As already mentioned in the introduction, existing taxonomies are not designed for application in the automotive sector and they typically focus on incident management. This means that they are not suitable for supporting a TARA or security testing. For that reason, we have decided to develop a taxonomy that allows the following three use cases to be covered:

1. Incident management,
2. threat analysis and risk assessment (TARA),
3. Security testing.

The first use case for which our taxonomy should be used is the analysis of incidents and vulnerabilities and the associated information gain which serves as a common language for security

incidents. It can be seen as a part of incident management and serves for classification of detected attacks and for forensic investigation [61]. In principle, information of the CERT/CC taxonomy is sufficient for this. However, we consider it to be useful to provide privileges achieved by an attack and security measures which have been overcome during an attack. The latter is particularly important for traceability of incidents and provides insight into how security mechanisms failed. In addition, by providing established vulnerability descriptions (such as CVE and common weakness enumeration (CWE)) it is possible to systematically derive security patches and therefore supporting patch management which can be a part of incident management.

The second use case we want to cover with our taxonomy is conducting a TARA [18]. Here, existing or planned systems are checked for possible attack scenarios. This results in a list of threats to the system. Since such a list can be very extensive, attack scenarios are prioritized in practice by assigning a risk value for occurrence and impact. This requires a detailed description of incidents and, in particular, knowledge of the affected assets (e.g., safety) and resulting consequences. In addition, by providing the underlying vulnerability of the threat, it is possible to determine to what extent a threat applies for other systems. This requires a detailed description of attack scenarios. The presented taxonomy can be used to provide a granular description and support the process of TARA. In combination with a database that records attacks already committed, the effort for TARA could even be reduced, as in that case the database could be reused for future projects.

A comprehensive description of incidents offers advantages for security testing and especially for the generation of test cases. Each attack carried out can be seen as a test case. In security testing, this is referred to as a penetration test [62], in which a targeted attempt is made to uncover security vulnerabilities through attacks in order to be able to make a statement about the security status of a system. The disadvantage of penetration tests is that they can only be carried out late in the development cycle. At that stage, it may not be possible to eliminate found vulnerabilities, or only with high effort. An alternative solution would be to carry out security tests during development in order to detect and eliminate possible vulnerabilities at an early stage. The detailed description of incidents serves to derive abstract test cases early, which can be realized as concrete test cases depending on the respective system structure (system architecture, system network, etc.).

To support all three use cases, we developed a classification model with a total of 23 categories containing several layers of abstraction. As Figure 2 shows, the abstraction is achieved by a hierarchical system, which starts with level 1 representing the highest level of abstraction up to level 3, which has the lowest level of abstraction. That would allow transferring attacks to different phases of the security development process if they were described with the taxonomy.

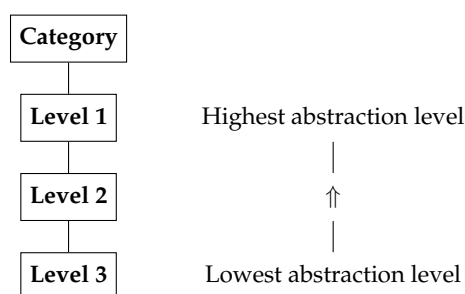


Figure 2. Description layers and their abstraction levels used in the taxonomy.

The high degree of abstraction in level 1 is particularly suitable for a comparable description of incidents for incident management as well as transferring attack paths to new systems and thus for early phases of the security development process like TARA. These typically use input information with a high degree of abstraction, since the technical design of the system is not yet known at this time. In later phases of the security development process, such as security testing, detailed descriptions of concrete attacks are required. This is provided by levels 2 and 3. Level 2 is also suitable for refining TARA results by concretize threats through more detailed descriptions. The individual

categories as well as their abstraction levels and relevance for the three use cases are described below. The application of our taxonomy on basis of exemplary incidents can be seen in Section 4. Subsequently we discuss quality aspects in Section 5. Criteria for an assessment such as uniqueness or completeness are presented and the extent to which our taxonomy fulfils these criteria is argued.

3.1. Description

To give a brief overview of the carried out attack, this category supplies a short description in continuous text. The description should include the effect, target and type of an attack. As shown in Figure 3 details are not listed here, as the category serves to provide a general overview of the attack.

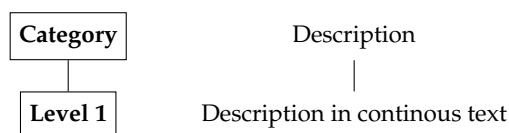


Figure 3. Short description of the attack.

This category includes only level 1, since it has the purpose of providing information about the attack in a short manner and is not absolutely necessary for a detailed TARA or security testing, which would require levels 2 or 3.

3.2. Reference

For reasons of traceability and reference, the publication source of an attack is described. Therefore, the authors and title of the reference are mentioned. If available, identification numbers (e.g., digital object identifier (DOI)) or a corresponding internet reference can be mentioned as illustrated in Figure 4.

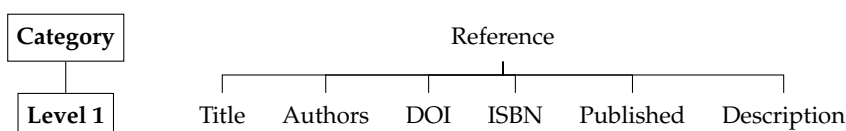


Figure 4. Reference of the described attack.

Since this category serves as a reference, only level 1 is addressed. The information provided by that reference is shown in the following categories.

3.3. Year

To provide information about an attack’s relevance, this section describes the year in which the attack was published, as shown in Figure 5.

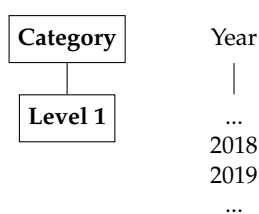


Figure 5. Year of the attack publication.

This category has no influence on TARA or security testing, since it only provides information about the year this attack was carried out.

3.4. Attack Class

This category describes the classification of an attack into a threat class. A problem with that classification are the different terms used in publications. For example, synonyms such as “sniffing”, “monitoring”, “listening”, “eavesdropping”, or “capturing” are used to describe the process of reading information from a component or communication system. The challenge in classifying a threat class is finding classes which provide a sufficiently generalized description, but show enough detail to comprehend the nature of an attack. For that reason, we have made a hierarchical classification into three classes as shown in Figure 6. Level 1 represents the spoofing, tampering, repudiation, information disclosure, denial of service, elevation of privilege (STRIDE) classification of Microsoft [63]. This model is used by Microsoft to identify threats as part of threat modeling. Since the STRIDE terminology is established in the security community, we use it to define our attack class. On level 2, we propose a classification into typical attack types takes place to specify a threat. On level 3, the concrete term of an attack is listed as a reference to the attack publication.

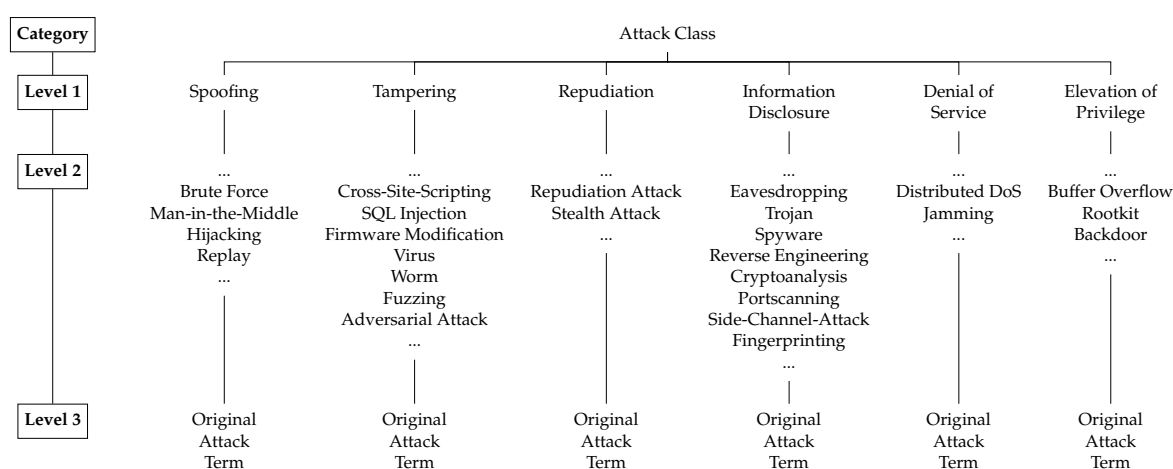


Figure 6. Classification of the attack based on spoofing, tampering, repudiation, information disclosure, denial of service, elevation of privilege (STRIDE) and common attack descriptions.

Level 1 can be used to associate known incidents with identified threats during TARA. For example, when a tampering threat has been identified, an analyst can check which specific attacks (level 2) come into consideration. This facilitates the determination occurrence probabilities for identified threats, since analysts can trace which concrete attacks have already been carried out in that category. Furthermore, levels 2 and 3 allow an estimation which attack techniques and thus which effort an attacker might have for attack execution. This information can still be used for security testing, since testers can include knowledge of the attack class for test case generation.

3.5. Attack Base

This category describes the area on which an incident is based. For that purpose, we propose the following distinction that is shown in Figure 7: wireless attack, diagnostic attack, bus attack, software attack, hardware attack, cryptographic attack, artificial intelligence attack and environment attack. Wireless attacks are primarily related to wireless communication links. Typically, this involves incidents aimed at connections from smartphones to telematics control devices or infotainment systems, but also keyless entry or navigation systems. Diagnostic attacks are based with diagnostic functions. Mechanisms such as security access [64] and flashing of new firmware concern this area. bus attacks relate to attacks on vehicle-internal networks. An attacker could control the actuators with controller area network (CAN) [65] messages for example. This category also includes direct physical connections between components. Software attacks refer directly to the software executed on a component.

Typically, these attacks aimed at modification or reverse engineering of a component’s firmware. That area includes attacks such as reading secret information from the software, infiltrating malicious code or command injection. Hardware attacks have an effect on physical components. Usually this concerns side-channel-attacks which aim to obtain information through measurements of the component’s physical quantities. Cryptographic attacks aim at bypassing or breaking cryptographic measures. An example is a cryptanalysis [66] of an immobilizer cipher. Artificial intelligence attacks refer to systems which use artificial intelligence algorithms [67]. These attacks typically use adversarial examples [68] to deceive algorithms such as person recognition. This is particularly relevant for autonomous driving, since these algorithms are applied and directly related to the safety of vehicles. Environment attacks describe manipulations of the environment to trigger unexpected behavior and are closely linked to artificial intelligence in autonomous driving. That area includes manipulations of traffic signs to deceive traffic sign recognition of autonomous vehicles, but influencing sensors such as camera or LiDAR [8] as well.

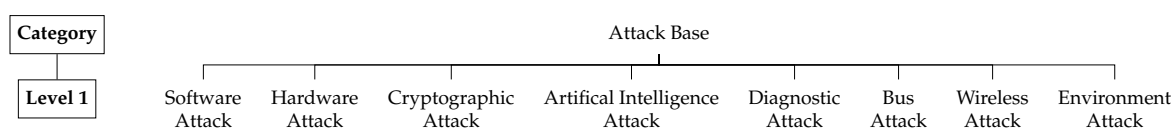


Figure 7. Base area of the described attack.

Level 1 elements allow analysts to make an initial assessment of attack complexity which is part of risk assessment. For example, if an attack is categorized as a hardware attack, it can be assumed an attacker requires physical access to the system and will most likely need tools to interact with hardware. On the other hand, wireless attacks would indicate an attacker does not require physical access and could potentially launch a remote attack. In addition, the seven subcategories can be used to identify a general test setup for a security test. For instance, environment attacks can be used to estimate that the environment around a test object must operate in a special manner (e.g., manipulated traffic signs).

3.6. Attack Type

In order to determine whether an attack is a simulation, an actually executed attack or a theoretical consideration, the type of incident is described in this category. A distinction is made between analysis, simulation and real attack as illustrated in Figure 8. The analysis is not an actual attack, but an investigation of a system for vulnerabilities and attack possibilities. In simulation, attacks were carried out on a simulative level. That includes attacking a system within a simulation environment. In a real attack, existing systems such as vehicles are attacked.

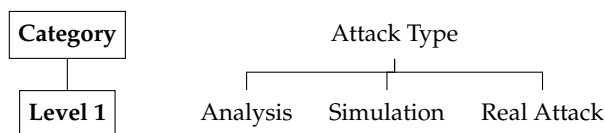


Figure 8. Classification of the attack type into analysis, simulation and real attack.

Although analysis and simulation do not include actually executed attacks they still provide useful information for conducting a TARA or security testing. Conceptual attacks can be understood as potential threats and hence they can be used as an input of a TARA. For testing purposes all elements of these category are relevant, since testing can already be performed on a simulation level.

3.7. Violated Security Property

Since every security attack implicitly violates a security property, this category addresses this area. For that reason, we chose common security properties as classification model that can be mapped to the

STRIDE threat model [69]. In the following security properties, shown in Figure 9, will be explained. Confidentiality describes the secrecy of data. This concerns personal and private data or information that is subject to data protection regulations and intellectual property. Generally, confidentiality is guaranteed by using encryption algorithms. integrity describes the correctness of information in a way it can not be altered by unauthorized persons. This is mostly achieved by checksums or message authentication codes (MACs) [66]. Availability describes permanent accessibility of resources over a long period. That means an attacker must not be able to prevent the provision of data or misappropriate information. Authenticity validates the identity of a sender from whom information is obtained. Non-repudiation describes the ability not to deny an action, such as sending or receiving a message. Authorization describes the ability to have privileges for performing certain actions. An example is the privilege to read critical information or to control safety-related components.

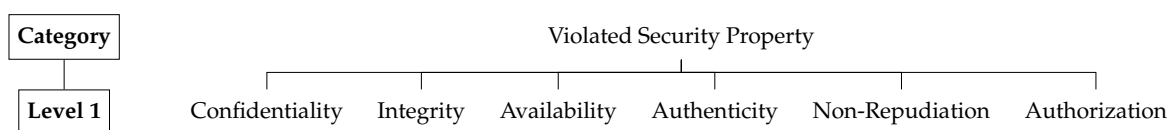


Figure 9. Specification of violated security properties which can be mapped to the attack class.

The knowledge about violated security properties is particularly useful for calculating risk metrics such as the common vulnerability scoring system (CVSS) [70] which is a rating method for vulnerabilities. Thus, violated properties can directly be transferred to the base score of CVSS, especially for confidentiality, integrity and availability. Furthermore, concrete countermeasures for security concepts can be derived. If we consider spoofing of a diagnostic message on the CAN [65] bus, the violated security property would be authenticity. So the measure to be implemented must ensure authenticity of messages which could be guaranteed by a keyed-hash message authentication code (HMAC) [66]. This category allows a direct derivation of high level security measures based on TARA results. An example of that process is given in [4].

3.8. Affected Asset

In order to identify objects of protection affected by an attack, this section is divided into four assets. We propose a distinction between safety, reliability, information security and financial assets as shown in Figure 10 because we see these four as the most relevant assets for the automotive area. Asset safety will be chosen if the incident has an influence on the actuators of a vehicle which influences driving physics like brakes, steering, or engine. Influencing these components can potentially affect the operational safety of a vehicle and endanger drivers, occupants and road users. Reliability describes vehicle functions which have no influence on operational safety. Information security includes assets which affect information values like personal data or firmware images. Financial assets mainly concern the theft of vehicles or vehicle components. Tuning or technical measures which influence the market value or warranty of a vehicle can be assigned to that asset as well as the reputation of a manufacturer.

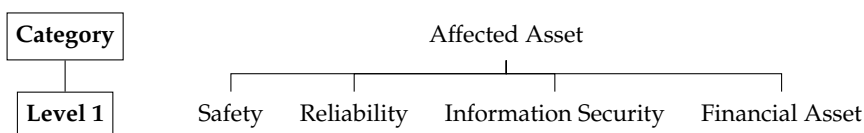


Figure 10. Description of affected assets.

The assignment of asset categories allows analysts to assess consequences of an attack. This is particularly useful for risk identification. Furthermore, with that classification an analyst is able to distinguish whether an attack violates the safety of a vehicle and can prioritize it with a higher risk value.

3.9. Vulnerability

An important information regarding a security attack is the vulnerability that made an attack possible. There are several approaches to describe vulnerabilities such as Seven Pernicious Kingdoms [71] or preliminary list of vulnerability examples for researchers (PLOVER) [72]. One concept that considers these approaches and many existing vulnerabilities taxonomies is the CWE [73]. This description model divides software security weaknesses into a hierarchical list model. CWE distinguishes following concepts: research concepts, development concepts and architectural concepts. architectural concepts subdivide vulnerabilities for classifying software architecture vulnerabilities. Development concepts classify vulnerabilities into categories which are often found in software development. We focus on the research concepts, as that model presents vulnerabilities and their interrelationships in an abstract and detailed way. Currently, the research concepts includes 806 CWE numbers and thus vulnerabilities. Since this is a high number of CWEs to choose from, we adapted their hierarchical structure and integrated it into our own hierarchical leveling. The most abstract level (level 1) represents high level vulnerability classes of the CWE research concept which is shown in Figure 11. The following classes are distinguished: CWE-682: incorrect calculation, CWE-118: incorrect access of indexable resource ('range error'), CWE-330: Use of insufficiently random values, CWE-435: improper interaction between multiple correctly-behaving entities, CWE-664: improper control of a resource through its lifetime, CWE-691: insufficient control flow management, CWE-693: protection mechanism failure, CWE-697: incorrect comparison, CWE-703: improper check or handling of exceptional conditions, CWE-707: improper enforcement of message or data structure, CWE-710: improper adherence to coding standards. These elements are subdivided into specific CWE numbers. By specifying the exact CWE number at level 2 of our vulnerability category, each attack vulnerability can be described in more detail. This provides a link to the CVE database that also assigns CWE numbers to the stored vulnerabilities. Due to that, we assign a CWE number to every attack that is described with our taxonomy. Finally, on level 3 the vulnerability is indicated as it was described in the attack publication. The level of detail can be as high as desired. An concrete example of this vulnerability classification is shown in Tables 1 and 2 in Section 4.

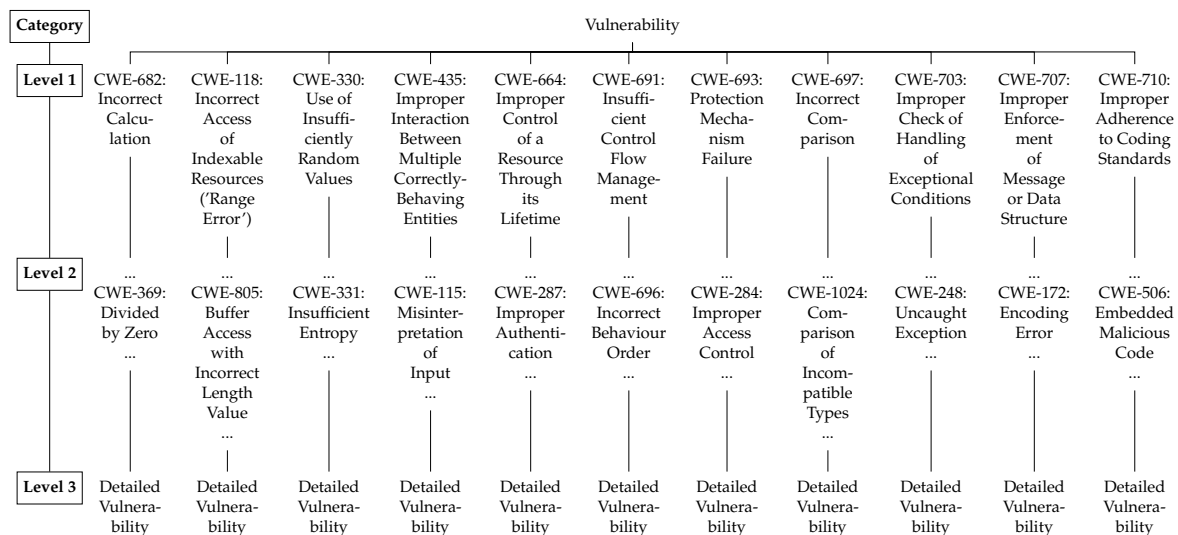


Figure 11. Vulnerabilities described with the common weakness enumeration (CWE) at different abstraction levels.

The knowledge of the vulnerability that made an incident possible can be used to support the process of TARA in order to avoid already known security vulnerabilities in newly developed systems. This concerns in particular the development of security measures to protect systems against attacks.

Information on vulnerabilities of a system is particularly relevant for testing. At this point a tester can use known vulnerabilities to test (attack) a system based on that information.

3.10. Interface

Each incident is characterized by an entry point which the attacker used to execute an attack. In this category, we describe these entry points. Since different technologies are used in vehicles for both internal and external communication, there are comparatively many interfaces. The following interfaces exist for external communication (remote) [17]: WLAN, Bluetooth, Cellular, global positioning system (GPS), digital video broadcasting-terrestrial (DVB-T)/digital video broadcasting-terrestrial, 2nd generation (DVB-T2), digital audio broadcasting (DAB), radio data system (RDS), tire pressure monitoring system (TPMS), radio frequency identification (RFID) and Infrared. The following interfaces exist for internal communication [74]: universal serial bus (USB), auxiliary (AUX), compact disc (CD), digital versatile disc (DVD), electronic control unit (ECU), Component, on-board diagnostics (OBD), CAN, controller area network flexible datarate (CAN FD), local interconnect network (LIN), media oriented system transport (MOST), desktop-bus (D-Bus), FlexRay, Ethernet, Camera, LiDAR, radio detection and ranging (Radar), Software. Due to space reasons, only a subset of interfaces is illustrated in Figure 12 (the complete list of interfaces is available at [37]).

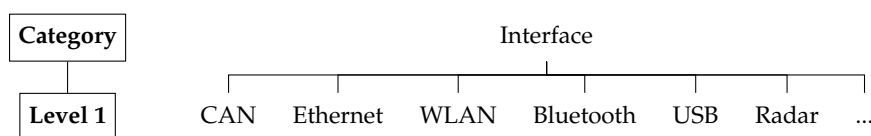


Figure 12. Interface used to communicate with a vehicle or system.

Behind each interface a certain technology (level 1) is located which an attacker must use to communicate with the attack target. Due to the fact that the interface technologies’ complexity varies widely, this category provides a valuable input for estimating the attack effort and thus for determining the feasibility of attacks. That is particularly important for risk assessment because estimating the effort of an attack is difficult and typically highly experience-based. In addition, this description provides valuable input for automating test case generation. With knowledge about the interface technology that is necessary for an attack, required driver layers (hardware abstraction layer (HAL) [75]) can be automatically derived and provided for a test case which can reduce the effort for test setups.

3.11. Consequence

The resulting consequences of a security attack are important when it comes to analyzing incidents. So in this category the attack’s effects on a vehicle or system are described (see Figure 13). The entries are written in continuous text. Due to various possibilities of effects an attack can have on a vehicle, it is hard to classify these effects especially when it comes to abstraction. This could be an aspect that should be addressed in the future. The description is made in such a way that the exact reaction of a vehicle or system is explained and potential consequences are mentioned.

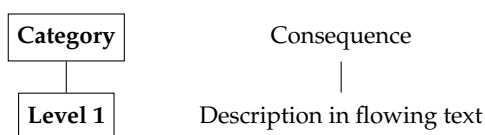


Figure 13. Consequence and impact of an attack.

Analysts can use the consequence description to refine potential damage to an attack target when conducting TARA. This category can be seen as a postcondition of the system when it comes to testing because precondition and postcondition are two essential states of a system during a test [76].

3.12. Attack Path

Many of the taxonomies presented in Section 2 describe incidents in an abstract way. Thus, valuable information about details of an attack are lost. We see an advantage in describing the steps of an incident as precisely as possible, since this information can be particularly relevant for the feasibility of attacks on other systems. Due to that, a detailed description of the exact procedure in multi-stage attacks should be achieved. Multi-stage attacks are described on basis of the presented taxonomy and broken down into its individual steps. In Figure 14 the structure of this description is illustrated.

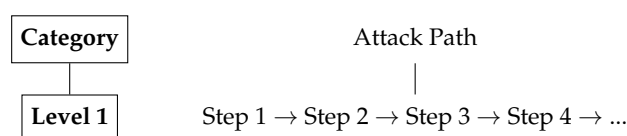


Figure 14. Specification of single-stage or multi-stage attack paths.

Since all multi-stage attacks are separated, it is possible to recombine individual steps. Attacks carried out by different sources can now be recombined and so far still unknown attack paths could be identified. In addition, separating incidents into several steps improves their transferability to new architectures. More precisely, it is now possible to convey only a part of an attack to a new architecture when a certain attack step is not feasible. This advantage is particularly relevant when carrying out a TARA, since the potential of attacks on a system is analyzed here. By separating an attack into its individual steps, a more precise statement can be made regarding the feasibility of an attack on a new system. Thus, the risk value of an attack can be determined more precisely. For security testing, valuable information can be taken out of this category. Knowledge of attack paths provides testers with information to find new paths based on a certain intermediate attack step. That allows higher test coverages during testing and could reveal new vulnerabilities.

3.13. Requirement

As already explained in Sections 1 and 2, there are different description models for incidents or threats. However, they hardly address necessary attack preconditions. For that reason, this category describes prerequisites, that had to be met, in order to carry out a corresponding attack. We propose the following five elements which are illustrated in Figure 15: required access/connection, required operational state, required system knowledge, required setup, organizational requirement. Required access/connection describes the type of vehicle access an attacker had to gain for carrying out attacks. Both wireless transmission paths and direct connections to bus systems or components are possible. Required operational state describes the state a vehicle has to be in at an incident event. That affects the condition of a vehicle and its subsystems. Necessary system information an attacker has to collect is described by required system knowledge. This applies to data which had to be read or obtained beforehand for a successful attack (e.g., identifying the vehicle identification number (VIN)). Required setup describes settings, configurations, simulations, etc. an attacker had to prepare. Organizational requirements describe organizational precautions an attacker must fulfil. This concerns the proximity to a target vehicle, or the attack time. Another example is the presence of a second attacker to execute an attack (like a two-thief attack to open a vehicle keyless [26]).

Since generally every attack can be interpreted as a successful penetration test, requirements for an attack can be considered as a test precondition. A direct mapping of our taxonomy for test execution is possible. Information about preconditions can be used to determine test configurations as well as test setups to bring a vehicle or system into the condition to be able to execute a test (attack). It provides valuable information for determining the probability of occurrence of an incident as it describes the necessary state for successful attack execution. Due to the fact that an incident presents itself as a realized threat, this information enhances the determination of threat probabilities during TARA. This would be particularly useful in determining the probabilities of occurrence if the attack

potential was included in the assessment, as proposed e.g., by e-safety vehicle intrusion protected applications (EVITA) [77]. We deem that a consistent description of necessary prerequisites can make risk assessment more consistent.

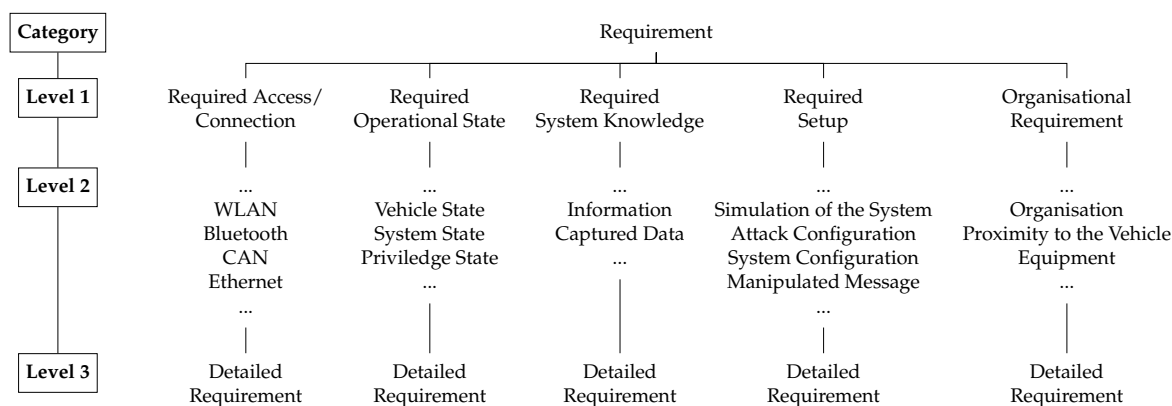


Figure 15. Requirements which were necessary to carry out the attack.

3.14. Restriction

Another important factor regarding successful incidents are restrictions making an attack execution more difficult. Due to that, we decided to add this category to our taxonomy which can be seen in Figure 16. We classify restrictions according to following elements: physical restriction, security measure, impact restriction, knowledge restriction, connection restriction, operational restriction. Physical restrictions describe physical influences, like range in wireless communication with a vehicle, that complicated an attack. Security measures are mechanisms already implemented in a vehicle. A well-known representative in the automotive sector is security access [64] used in vehicle diagnostics. Impact restrictions limit effects of an attack, for example, attacks which were only possible up to a certain vehicle speed or steering angle [14]. Knowledge restrictions describe information-related restrictions. Connection restrictions refer to vehicle connections. An example is the handshake process for establishing a WLAN connection. Operational restrictions limit an incident by the operational state of a vehicle or its subsystems. A real example is an attack in which a malicious smartphone app was developed that triggered transmissions of CAN messages as soon as the driver connects his smartphone to the vehicle. The need for the app to be distributed and installed by a driver on his smartphone is an operational restriction [17].

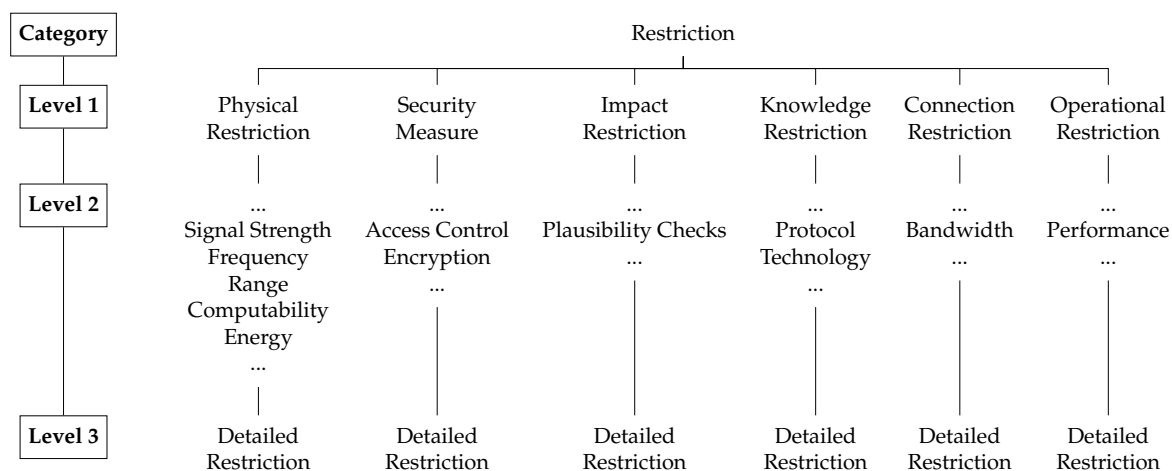


Figure 16. Restrictions which prevent the full impact of an attack.

In this category the sub-item security measure is of particular value because it provides information why certain security features could be defeated and thus no sufficient protection was ensured. Based on that knowledge, improvements can be applied or security features can be replaced by more secure solutions. A concrete example is the security access which is implemented with a Seed and Key procedure. Although that concept violates Kerckhoff's principle [78], it is still typically used in modern vehicles. However, if a weakness of the concept was known during development, a different implementation would be chosen. In addition, understanding restrictions provides a more detailed assessment of attack complexity, which is important for risk assessment of threats. Estimating these values is not an easy task and experience-based. Therefore, they vary greatly between different analysts which in turn negatively affects the comparability of risk values. The category counteracts this by providing each analyst with the same detailed information base and mitigates situations of a false assessment due to a lack of information. It can support security testing by providing information about the state in which a test object has to be in for test execution. Additionally, this category supports testers by supplying information about security measures which have to be tested or overcome in order to test an object.

3.15. Attack Level

In case of attacks on vehicles, attack paths can be extensive. An example is access to a vehicle via a remote interface and an actuator accessed through several vehicle-internal communication channels and ECUs (as in [15]). In such incidents, individual attack steps are distributed over several levels which should be taken into account. So this category describes the level on which an attack took place. As shown in Figure 17 we distinguish between remote, local network, local software and local physical. With remote, the attack is primarily carried out via wireless connection to a component in a vehicle. Local network incidents occur via physical connection to the internal vehicle network (e.g., via OBD). In a Local Physical attack, the attacker has direct physical access to a component such as an ECU. Local software attacks affect the software running on a component as well as extracted firmware images.

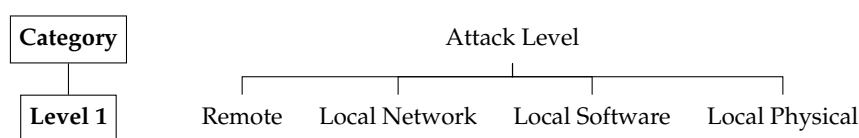


Figure 17. Level on which the attack was carried out.

Knowledge about the attack level helps to determine the incident potential and thus to assess risk during a TARA. Furthermore, when planning security tests, resources can be allocated at an early stage. For example, it makes sense to plan all test cases on the local physical level together because all cases require a physical connection to a target.

3.16. Acquired Privileges

In order to carry out an attack, there are privileges the attacker has to obtain. Since a vehicle is a cyber-physical system [23], IT privilege models cannot be transferred directly. Most automotive ECUs have no operating system that is comparable to the IT area, so there is no root or similar user authorization on these ECUs. From an attacker's point of view, there are still privileges to obtain before execution of certain operations is possible. For this reason, we look at permission levels from an attacker's point of view. As a result, different levels must be considered for vehicles. At first there has to be a communication path from an attacker to the vehicle. Furthermore, the component an attacker is communicating with must be taken into account. In order to do so, mechanisms such as access control or encryption has to be considered. By taking these requirements into account we distinguish two points of view: the physical view and the informational view. With the physical point of view, communication to a component with a communication medium and associated privileges, in general, are always possible as soon as a connection to the corresponding medium has been established. As an

example, communication with the CAN bus can be mentioned. As soon as an attacker has access to CAN, she achieved the privilege to send and read messages. However, an attacker cannot necessarily understand read message content she wants to send by achieving that privilege. We consider this knowledge as a part of the informational point of view. So an attacker has the privilege to read and send physically and understand data contents as well as any protocols or security measures. Thus, security measures can be covered. The following privileges management we propose to describe achieved privileges by an attacker takes the informational point of view which includes the physical point of view implicitly. The privileges management is divided into following five privileges levels which are illustrated in Figure 18: read/write (functional communication link), execute (functional component), read (functional component), write (functional component), full control (functional component). Read/write (functional communication link) describes the privilege to communicate with a vehicle via a communication interface. This means that read and write accesses to the communication medium are possible. That privilege covers the level for communication with a vehicle and represents the lowest privilege that an attacker must obtain in order to execute an attack. Execute (functional component) describes the initiation of a function on a component. This can be done by sending a message via the communication medium to an ECU. An example is sending a diagnostic message to the CAN bus to trigger a diagnostic function like reading diagnostic fault memory. Read (functional component) describes the privilege to read information from a component. That includes diagnostic data as well as secret information such as secret keys for security measures. Write (functional component) describes the ability to write or change data on a component. This can be achieved by initiating a calibration update which writes specific information to an ECU. Full control (functional component) is comparable to root privileges in user administration. That privilege grants an attacker full control of a component. Usually this is achieved by flashing a malicious firmware on a component.

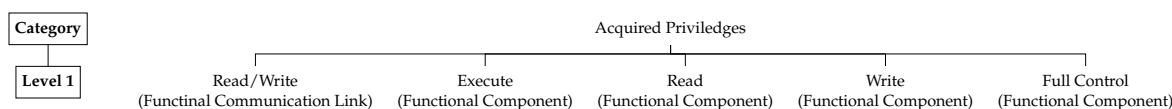


Figure 18. Privileges an attacker needs to acquire for a successful attack execution.

The description as functional component or functional communication link enables our proposed privilege management to be used in combination with previously presented attack levels. For example, on remote level, read/write (functional communication link) privilege describes the communication technology used and functional component privileges describe the target communication component. This scheme can be used across all attack levels to describe privileges at different levels of an attack. Achieved privileges are especially helpful for the generation of new attack paths. In principle, attacks could be combined to a path if necessary privileges of an attack level matched privileges of the previous attack level. Based on that concept, we are currently developing a tool that uses this category to automatically generate attack paths in vehicle architectures and use them for risk assessment and security testing.

3.17. Vehicle

In order to identify attacked vehicles, this category describes the vehicle information in detail. The vehicle manufacturer, model, construction year, type and additional information that may be relevant for an attack can be included as shown in Figure 19.

Knowledge about the vehicle can help car manufacturers during TARA to determine if an incident is transferable to other vehicles of their fleet. With this knowledge, conclusions can be drawn about the network architecture which can represent important information on transferability in security testing, as there is potential for reusing test cases.

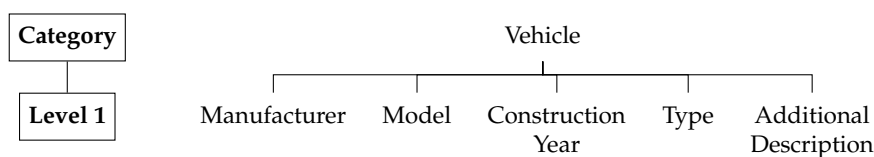


Figure 19. Description of an attacked vehicle.

3.18. Component

The knowledge about the target component is an essential information. As illustrated in Figure 20 we distinguish three types of components: ECU, sensor and actuator. Since vehicle manufacturers often use different designations for in-vehicle components, we provide different entries for each category. For ECUs components such as engine ECU or airbag ECU are provided. The manufacturer-specific component designations or designations mentioned in attack publications are appended to the selected component. An example entry could therefore have the following form: ECU (level 1), engine ECU (level 2), engine control unit version 1.09 (Level 3).

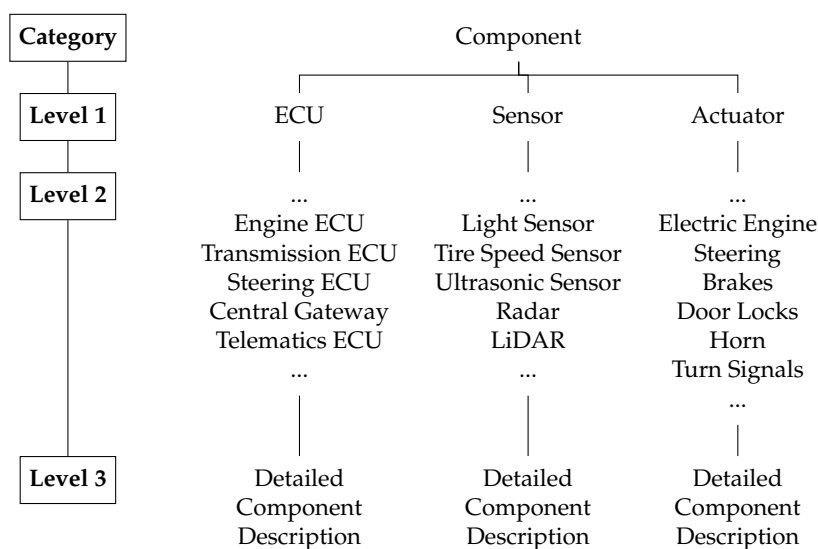


Figure 20. Specification of the target component.

This category provides valuable artifacts for threat analysis. An initial decision whether an incident can potentially affect the safety of a vehicle is required. That will be the case if an attack has an actuator as target because actuators can actively influence a vehicle’s driving physics. So a tester or an analyst can verify whether attacks were performed on similar systems in the past to make new systems more secure against these attacks.

3.19. Tool

In order to carry out an attack, the attacker requires special equipment for communication with a vehicle and the actual execution of an attack. As shown in Figure 21 we distinguish these tools between security tools, software tools, hardware tools, sensing tools, measurement tools and wireless tools. Software tools include libraries, drivers, simulation environments and similar which were necessary to execute an attack. Hardware tools contain the physical hardware used for the attack. Wireless tools mainly include wireless communication equipment to communicate over the air. Measurement tools contain equipment for physical measurements like oscilloscope or logic analyzer. Sensing tools describe sensors which were necessary to execute the attack. Security tools are a special section in this category, since these tools could be software tools or hardware tools, for example. We determine that security tools include all tools whose purpose is related to a security process like attacking, investigating or

defending a system. Therefore, we used the tool classification that offensive security uses for Kali Linux [79].

The knowledge of used tools and required resources can help to determine the attack potential and supports the risk assessment during TARA. It is possible to estimate necessary resources for testing. This allows reducing testing effort by grouping tests which address the same tool category.

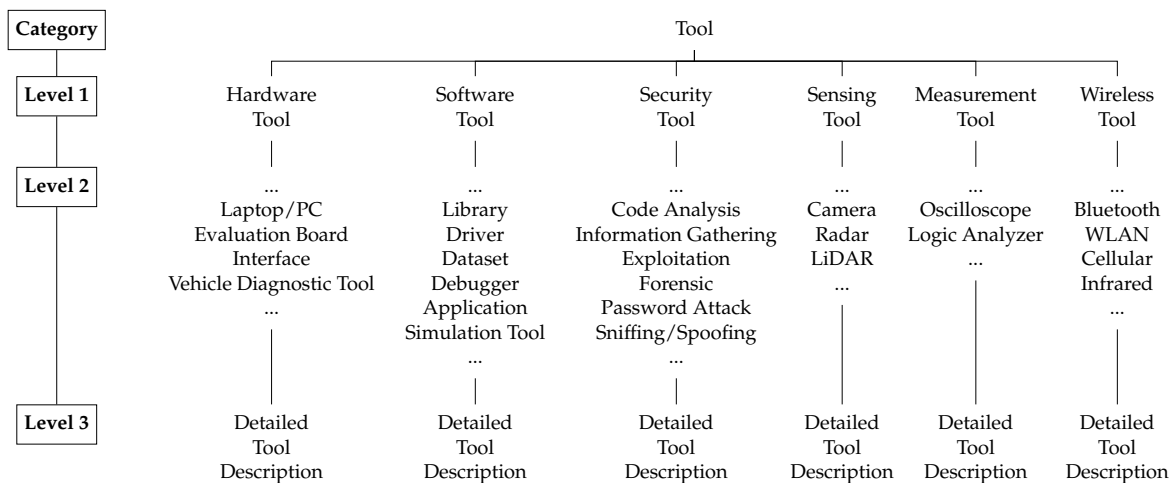


Figure 21. Specification of necessary attacking tools.

3.20. Attack Motivation

The attacker’s motivation to execute an attack is an information that can be used to determine the risk. We consider the description from Hoppe et al. [59] as reasonable. The authors extended the objective category of CERT by adding the elements Security Evaluation and tuning to the already existing elements challenge status/thrill, political gain, financial gain and damage. Since that taxonomy covers use cases of the automotive sector, we used the classification of Hoppe et al. for attacker’s motivation as shown in Figure 22.

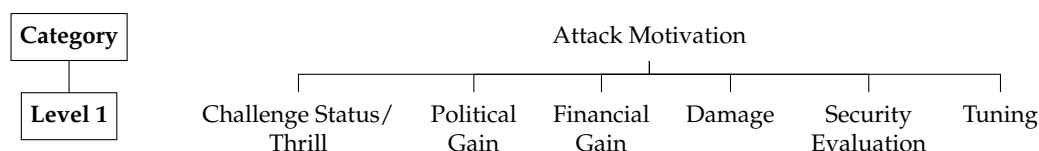


Figure 22. Attacker motivation or objective to attack a vehicle or system.

Although the inclusion of attacker motivation in risk assessment can be considered critical due to its high subjectivity, we have decided to support it in our taxonomy due to its usage in TARA methods.

3.21. Entry in Vulnerability Database

In case of already existing entries of automotive vulnerabilities in databases, this category is used to specify that database. One example of a widely used vulnerability database is the national vulnerability database (NVD) [38] which uses the common vulnerabilities and exposures enumeration (CVE) [39] list. Further examples are Rapid7—vulnerability and exploit database [45] used by the metasploit framework [80], CERT/CC Vulnerability notes database [81], and packet storm database [43]. In addition, there are other databases in which vulnerabilities can be identified. We already mentioned them in Section 2. To our knowledge, automotive vulnerabilities are only represented in the CVE list. The databases shown in Figure 23 are only a subset of existing ones.

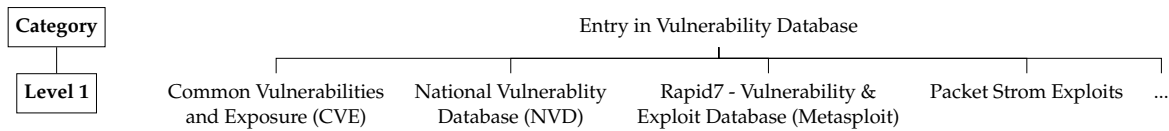


Figure 23. Specification of vulnerability databases.

The information whether a vulnerability or an incident is mentioned in a vulnerability database serves as a reference to where the vulnerability is noted and how it is described. A benefit for TARA or security testing will only arise if many automotive vulnerabilities are registered into a database in a form which provides relevant information for these processes.

3.22. Rating

In the context of TARA, attacks are checked for their probability of occurrence and severity when they are carried out in a system. On the basis of that information, a risk value is determined and assigned to the respective attack. That process is implemented in this category and an incident or vulnerability is evaluated according to its severity. This allows an attack to be used directly for threat analysis and risk assessment. Such ratings allow attacks to be prioritized for their relevance and thus serve as prioritized test cases. We currently use the CVSS [70], as shown in Figure 24, to evaluate attacks as that evaluation scheme is widely used and uniform. In addition, elements for determining that value can be mapped directly to categories of our taxonomy. For example, the CVSS category attack vector can be mapped with the four entries of our taxonomy category attack level. However, instead of CVSS other ratings systems could be used for this category.

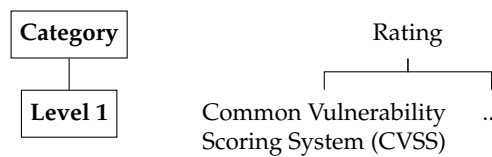


Figure 24. Rating of the attack’s severity.

The CVSS value is calculated from several sub-values and shows the relevance of a vulnerability. The advantage of using such risk values is their uniformity. This mitigates situations where different risk values are assigned for a vulnerability by different analysts. That could improve the comparability of TARA results.

3.23. Exploitability

In addition to the rating value explained above, exploitability can be used as an assessment of the attack’s relevance when considered separately. The exploitability represents a metric for the probability of occurrence, independent from other metrics such as severity. As shown in Figure 25, we use the exploitability value which results as a partial value within calculation of the CVSS value.

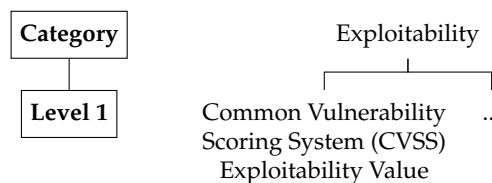


Figure 25. Exploitability rating of an attack or vulnerability.

We consider that the exploitability value can make a valuable contribution to determining the probability of occurrence of threats or attacks. It can be seen as a substitute for attack potential and

is free of subjective influences like the attacker's motivation. It is therefore a reasonable alternative to complex metrics such as those used by EVITA. In addition, this value is defined once and can be consistently used afterwards. We are planning to use the exploitability value in combination with the severity value of an automotive safety integrity level (ASIL) of the ISO 26262 [82] which represents a metric of functional safety to determine the risk for safety-affecting threats.

4. Example

In this section, two exemplary incidents are shown which are classified according to our taxonomy. These two examples are used to demonstrate its usability. The first example shows a comparatively simple attack which is executed over one stage. The second incident is a multi-stage attack consisting of several stages with different characteristics. In Table 1 the first attack is listed. It was carried out in the work of Koscher et al. [16]. For space reasons, categories that only cover level 1 are displayed in one column across all three levels in Tables 1 and 2. This means that these elements have only one degree of abstraction (level 1), but it extends over all levels as global information.

The incident describes deactivation of the ECU communication over CAN by using a standard diagnostic function published in the year 2010 in the publication experimental security analysis of a modern automobile by Koscher et al. The attack is classified as a denial of service (category: attack class) because communications of the ECUs via CAN were blocked. Since this attack class was not described in detail in the publication, level 2 and level 3 get the entries none. The incident was primarily triggered by a diagnostic function, so the attack is a diagnostic attack (category: attack base) and since it was actually executed, we consider it as a real attack (category: attack type). Furthermore, the attack violated the availability (category: violated security property) due to the denial of service. Since the incident did not trigger any safety-critical functions and neither information theft nor financial benefits were obtained, the asset reliability (category: affected asset) was affected. To execute the attack, a connection to OBD (category: interface) had to be established. The vulnerability that enabled the attack affects the improper control of a resource through its lifetime (category: vulnerability). The CWE number CWE-284, which stands for the vulnerability improper access control, was assigned to this incident in order to specify the vulnerability on level 2. Level 3 describes standard is not met because "disable CAN communication" command must not be executed when vehicle is in motion in detail. Category attack path only includes one stage: sending the standard command for disabling the CAN communication via OBD. The prerequisite for that is required access/connection (category: requirement) via OBD. There were no restrictions for that attack, which is why in this category levels 1–3 get the entries none (category: restriction). Since the incident is related to the vehicle's network, local network (category: attack level) is chosen. Koscher et al. were able to trigger a function on a corresponding ECU and thus possessed execute (functional component) privileges (category: acquired privileges) which implicitly contains read/write (functional communication link), since communication via CAN already existed before. The attack ensured that communication between different ECUs (category: component) was no longer possible. For space reasons, only the engine ECU and electronic brake ECU as well as their exact designations (level 2) are shown in Table 1. The brand of the attacked vehicle is not known, so there is the entry vehicle brand not known in category vehicle. In the tool category tools used for attacks are shown. Level 1 includes a general description of the tool (e.g., hardware tool), in level 2 the special variant (interface) and in level 3 a detailed designation (CAN-to-USB-converter) is shown. Since the mentioned publication is a research work, the attack motivation was security evaluation (category: attack motivation). So far there is no entry in the vulnerability database. The incident was evaluated according to CVSS, whereby a rating of 7.4 (category: rating) was achieved. From this rating the value 2.84 for category exploitability could be extracted.

Table 1. Example of a single-stage attack described by our taxonomy.

Category	Level 1	Level 2	Level 3
Description	Deactivation of the ECU communication over CAN by using a standard diagnostic function		
Reference	Experimental Security Analysis of a Modern Automobile (Koscher et al.)		
Year	2010		
Attack Class	Denial of Service	None	None
Attack Base	Diagnostic Attack		
Attack Type	Real Attack		
Violated Security Property	Availability		
Affected Asset	Reliability		
Vulnerability	CWE-664: Improper Control of a Resource Through its Lifetime	CWE-284: Improper Access Control	Standard is not met because “disable CAN communication” command must not be executed when vehicle is in motion
Interface	OBD		
Consequence	ECU communication is disabled while driving the vehicle		
Attack Path	Sending the standard command for disabling the CAN communication via OBD		
Requirement	Required Access/Connection	OBD	None
Restriction	None	None	None
Attack Level	Local Network		
Acquired Privileges	Execute (Functional Component)		
Vehicle	Vehicle brand not known		
Component	Engine ECU	Electronic Brake ECU	Engine Control Module
	Electronic Brake ECU	Electronic Brake Control Module	None

Tool	Hardware Tool	Interface	CAN-to-USB-Converter
	Hardware Tool	Laptop/PC	None
	Measurement Tool	Oscilloscope	None
	Hardware Tool	Evaluation Board	Atmel AVR-CAN
	Hardware Tool	Interface	CANCapture ECOM Cable
	Software Tool	Communication Tool	CARSHARK
	Security Testing Tool	Reverse Engineering Tool	IDA Pro
Attack Motivation	Security Evaluation		
Entry in Vulnerability Database	None		
Rating	CVSS: 7.4		
Exploitability	CVSS Exploitability: 2.84		

The second attack, presented in Table 2 and executed by Miller and Valasek [14], is a multi-stage attack. We do not describe all categories, only important ones like attack path.

Table 2. Example of a multi-stage attack described by our taxonomy.

Category	Level 1	Level 2	Level 3
Description	Unauthorized flashing of malicious code on the engine ECU by using the diagnostic reprogramming routine		
Reference	Adventures in Automotive Networks and Control Units (C. Valasek et al.)		
Year	2013		
Attack Class	Tampering	Firmware Modification	None
Attack Base	Diagnostic Attack		
Attack Type	Real Attack		
Violated Security Property	Integrity		
Affected Asset	Information Security		
Vulnerability	CWE-693: Protection Mechanism Failure	CWE-287: Improper Authentication	Unauthorized reprogramming possible
Interface	OBD		
Consequence	Flashing of malicious code on ECU		
Attack Path	Downloading a new calibration update for ECU from manufacturer and Reverse Engineering of the Toyota Update Calibration Wizard (CUW). Monitoring the update process. Reverse Engineering update algorithm for calibration updates. Modification of calibration update. Reflashing of malicious update.		
Requirement	Required Access/Connection	OBD	None
Restriction	Security Feature	Access Control	Security Layer which is tied to the Calibration Version and allows only one time overwriting
Attack Level	Local Network		
Acquired Privileges	Full Control (Functional Component)		
Vehicle	Toyota Prius (Year of Construction: 2010)		
Component	Engine ECU	Engine Control Module	2 CPUs, NEC v850, Renesas M16/C
Tool	Software Tool	Vehicle Diagnostic Software	Toyota Calibration Update Wizard (CUW)
	Hardware Tool	Interface	J2534 PassThru Device (CarDAQPlus)
	Hardware Tool	Interface	ECOM cable
	Hardware Tool	Laptop/PC	Windows PC
	Software Tool	Communication Tool	EcomCat Application
Attack Motivation	Security Evaluation		
Entry in Vulnerability Database	None		
Rating	CVSS: 6.8		
Exploitability	CVSS Exploitability: 1.62		

The attack consisted of unauthorized flashing of malicious code on the engine ECU by using the diagnostic reprogramming routine that took place via OBD (category: interface). Flashing of malicious code on ECU could be reached as consequence which violated the integrity (category: violated security property) and concerned information security (category: affected asset). We chose that incident as an example because it has several stages in category attack path, which are explained below. The first step was downloading a new calibration update for ECU from manufacturer and reverse engineering of the

Toyota update calibration wizard (CUW). This activity can be performed at any time and independently of the vehicle. The aim of this step is information gain about the target vehicle and its functionality in order to find possible weak points. Miller and Valasek pursued the goal to flash an ECU with malicious code. Therefore the next step included monitoring the update process to get information about the update routine. The third step, reverse engineering update algorithm for calibration updates, was based on information obtained before. The researchers retrieved all information they needed to flash calibration updates. The fourth step consisted of modification of calibration update and in the last step reflashing of malicious update. Splitting an incident into its individual attack steps has several advantages. A detailed description of the attack and carried out activities is given. Furthermore, individual attack steps can have properties which differ from the overall attack. When considering the incident as shown in Table 2, the violated security property can be described with integrity, since a modified update was flashed on ECU. Considering the attack path's first step separately: downloading a new calibration update for ECU from manufacturer and reverse engineering of the Toyota update calibration wizard (CUW), this step does not violate integrity, but confidentiality. Furthermore, requirement: required access/connection (level 1) and OBD (level 2) would not be necessary for that step. If individual attack steps are described with our taxonomy, different properties within an incident can be represented. Thus, more exact information for TARA as well as possible testing steps could be pointed out. As a further advantage, new attack paths can be found. Considering the first attack step again, we can see that the information gained from reverse engineering could have led to other attacks in addition to the malicious update. This was also done in the publication by Miller and Valasek [14], since reverse engineering the Toyota calibration update wizard (CUW) made it possible to identify diagnostic messages with which the researchers could control various actuators. Finding new attack paths is therefore a great advantage for security development, as these paths may not have been considered before and thus not secured. These information can also be used to identify possible places where security mechanisms can be introduced to secure several attack paths at once. Furthermore, activities can be identified that can be executed separately from the actual incident. For example, all reverse engineering activities illustrated in Table 2 can be performed offline from the vehicle. The information gain of these steps can help attackers to execute further attacks on a car. Such activities can occur in the middle of the attack path, i.e., an attacker establishes a connection to a vehicle and reads the firmware of an ECU. In order to understand the contents of the firmware, reverse engineering or disassembling would take place at this point. That process could be carried out by an attacker at any time and without a vehicle connection. If the attacker succeeds in extracting information from firmware, she can reconnect to the vehicle to use the information for an incident. In this case, the attack path would be interrupted until information gathering is complete and continued at a later stage.

5. Discussion

In this section, a critical analysis of our presented approach takes place and possible discussion points are taken up. It should be mentioned that our taxonomy has some disadvantages besides the advantages described above. A major drawback is the complexity caused by the high number of categories and their different hierarchical levels. For a single attack, the classification effort is manageable, but for a large number of attacks, the time expense rises significantly. In order to address that problem, we are developing a software tool which makes it possible to support the classification of attacks by providing a graphical interface for each category in which the corresponding possible entries can be selected. The entered attacks will be displayed in tabular form as well as a listing. In Section 6 we explain the planned functionality of our tool in more detail. In our opinion, the advantages resulting from our taxonomy predominate high effort disadvantages. The initial effort to describe many attacks is high, but these incidents can be reused in future, saving time in the long run. This will particularly be the case if attacks are stored within a database (vulnerability database), as it would provide a fixed knowledge base which could be used at any time to develop security concepts. We would also argue

that the complexity is reduced significantly when different stakeholders need different information. For example, when a stakeholder only needs information from level 1, the detailed information from level 2 and level 3 can be neglected. In this case, the effort a stakeholder would have to put in for an incident description, is comparable to other taxonomies like CERT's.

Another point of discussion is an evaluation of our approach. We applied our taxonomy to several published incidents. A collection of published attacks already made by the Karlsruhe University [9,83] served as a foundation. This collection was extended by further automotive attacks and classified by our approach. A table with a total of 162 attacks from 74 publications was created [37]. By consideration of a multi-stage attack's single steps, as described in Section 4, there are even 413 attacks [37], which we could classify. In addition to this internal assessment, we are planning a further evaluation in the future. Currently there are publications of executed automotive attacks which are not yet entered and classified in our collection. After finishing the tool described in the previous segment, these publications will be classified as an evaluation of security engineers using our taxonomy and will be added to the incident collection.

As further points of discussion the following six criteria are considered which were presented by John D. Howard in his dissertation [57]. Howard explained that a taxonomy should be mutually exclusive, exhaustive, unambiguous, repeatable, accepted and useful. So at this point we argue whether our approach meets these criteria.

Mutually Exclusive means that there should be no overlapping between possible entries within a category. Since automotive security attacks are often specified in varying degrees of detail in publications, classifications within a category are not exclusive. This is not a general problem of our taxonomy, but of the available information used to classify it. However, situations can occur where information can be assigned to several elements of a single category. This represents a general problem in security and has already been discussed in various publications [57,84]. Weber et al. [48] have also recognized that problem. The authors argue that it is a characteristic of flaws, if they can be assigned to several classified elements. Furthermore, the authors argue that the overarching sense of a taxonomy lies in usability and with semantic correctness under certain circumstances valuable information can get lost. We agree with their argumentation and would like to point out that our proposed classification scheme should above all support the presented use cases as an information base which supports the development and test process of security development.

Exhaustiveness is a general problem in the area of security, since no statement can be made as to whether all incidents or possible attack paths are known. Therefore, there is basically no completeness over all attack possibilities. We can only make the statement that we are exhaustive over the automotive attacks we have classified in our incident table. It is difficult to assess completeness, as only few automotive incidents have been published compared to IT where more incidents are public. An example is the dissertation of Howard [57] in which 4299 internet incidents were available. In this respect, a larger database could provide greater assurance regarding completeness.

The criteria of unambiguousness and repeatability can only be checked during an evaluation, since several parties would work with our classification and show its repeatability. In principle, that criterion can only be verified in practice anyway. However, we address the criterion of unambiguousness by splitting attacks into their single attack steps. Splitting attacks lead to a significantly better unambiguousness, since single attack steps break down the full attack which could be classified into several category elements. The criterion of acceptance can only be achieved by establishing our taxonomy in the security community. Due to this we want to introduce it already at this point in time, so we could get feedback and criticism already now.

We mentioned the usefulness of our taxonomy in Section 3 by describing three use cases we want to address with our approach in order to support security development in the automotive domain. We believe that a description of automotive security incidents in such a way can provide valuable information to security development and testing processes.

Finally, another criterion should be considered, which is subjectivity. When using taxonomies, there is a risk the person doing the classification will have to select several entries that exist within each category. However, that process is subjective and depends on the person in question. Remedy is provided by defined criteria on basis of which an entry is selected. In Section 3 the individual entries were described in order to support users of the taxonomy. This aspect should be part of the planned evaluation, since the applicability of our classification will be determined there. The problem of subjectivity applies in principle to every taxonomy. This occurs above all with categories like attacker motivation, since no criteria can be specified, in which situation an entry must be selected. However, a fundamental problem of security is reflected. For example, the process of TARA is also subjective. To reduce subjectivity as much as possible, we have limited ourselves to using established security description models such as STRIDE which have already been used for years in the area of security.

6. Conclusions and Future Work

In this work, we presented a taxonomy to describe automotive security attacks. The classification of attacks into categories of different detail levels enables using our taxonomy for TARA and security testing and provides a uniform description for security developers and testers. We have classified our collection of 162 published security incidents as well as the splitted version with 413 multi-stage attacks on vehicles based on this taxonomy and made it available online [37] in order to provide a public collection of incidents. By using description patterns such as STRIDE, which are already established in the security field, we hope to have created a common language for information exchange. This could improve cooperation between different stakeholders of the automotive domain by using this mutual language and collection of knowledge. We hope that our taxonomy will be accepted by the automotive security community in order to support developing security concepts for vehicles. We would also appreciate feedback and constructive criticism from the security community.

For future work, we plan to extend our attack collection by further incidents to expand our information base. In one project, our colleagues at the Karlsruhe University of Applied Sciences [21] are working on developing a vulnerability database for automotive security vulnerabilities. Our collected attacks and their classification can serve as an input which can be mapped in such a database. Another area we currently address is the development of a tool that can be used to classify attacks according. Since our taxonomy has comparatively many categories, which themselves have several classification elements, the tool improves the applicability, in particular by a high number of published incidents. In addition, extensions are to be made within the tool to enable the application of our taxonomy for TARA and security testing. With the completion of our tool, an evaluation will be carried out in order to evaluate the criteria mentioned in Section 5 as well as their applicability and acceptance.

Author Contributions: Conceptualization, F.S. and J.D.; data curation, F.S. and J.D.; funding acquisition, R.K.; methodology, F.S. and J.D.; project administration, R.K.; supervision, R.K.; writing—original draft, F.S. and J.D.; Writing—review and editing, F.S., J.D. and R.K. All authors have read and approved the final manuscript.

Funding: This research received no external funding.

Acknowledgments: We thank the reviewers for their inspiring comments.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Maurer, M.; Gerdes, J.C.; Lenz, B.; Winner, H. *Autonomes Fahren: Technische, Rechtliche und Gesellschaftliche Aspekte*; Springer-Verlag: Berlin, Germany, 2015.
2. IEEE. *IEEE Std 802.11ak-2018 (Amendment to IEEE Std 802.11(TM)-2016 as Amended by IEEE Std 802.11ai(TM)-2016, IEEE Std 802.11ah(TM)-2016, and IEEE Std 802.11aj(TM)-2018): IEEE Standard for Information Technology-Telecommunications and Information Exchange Between*; IEEE: Piscataway, NJ, USA, 2018.
3. Bluetooth Special Interest Group. Bluetooth Core Specification v5.0, 2018. Available online: <https://www.bluetooth.com/specifications/bluetooth-core-specification> (accessed on 28 March 2019).

4. Dürrwang, J.; Braun, J.; Rumez, M.; Kriesten, R.; Pretschner, A. Enhancement of Automotive Penetration Testing with Threat Analyses Results. *SAE Int. J. Transp. Cybersecur. Priv.* **2018**, *1*, 91–112. [CrossRef]
5. Francillon, A.; Danev, B.; Capkun, S. Relay attacks on passive keyless entry and start systems in modern cars. In Proceedings of the Network and Distributed System Security Symposium (NDSS), San Diego, CA, USA, 6–9 February 2011.
6. Keen Lab. Experimental Security Assessment of BMW Cars: A Summary Report. 2017. Available online: https://keenlab.tencent.com/en/whitepapers/Experimental_Security_Assessment_of_BMW_Cars_by_KeenLab.pdf (accessed on 17 April 2019).
7. Nilsson, D.K.; Larson, U.E.; Picasso, F.; Jonsson, E. A First Simulation of Attacks in the Automotive Network Communications Protocol FlexRay. In *Proceedings of the International Workshop on Computational Intelligence in Security for Information Systems CISIS'08*; Corchado, E., Zunino, R., Gastaldo, P., Herrero, Á., Eds.; Springer: Berlin/Heidelberg, Germany, 2009; Volume 53, pp. 84–91.
8. Petit, J.; Stottelaar, B.; Feiri, M.; Kargl, F. Remote attacks on automated vehicles sensors: Experiments on camera and lidar. *Black Hat Europe* **2015**, *11*, 2015.
9. Ring, M.; Dürrwang, J.; Sommer, F.; Kriesten, R. Survey on vehicular attacks - building a vulnerability database. In Proceedings of the 2015 IEEE International Conference on Vehicular Electronics and Safety (ICVES), Yokohama, Japan, 5–7 November 2015; pp. 208–212.
10. Spaar, D. Auto, öffne dich. *Sicherheitslücken bei BMWs ConnectedDrive C* **2015**, *5*, 15.
11. Verdult, R.; Garcia, F.D.; Balasch, J. Gone in 360 seconds: Hijacking with Hitag2. In Proceedings of the 21st 5USENIX6 Security Symposium (5USENIX6 Security 12), Bellevue, WA, USA, 8–10 August 2012; pp. 237–252.
12. Verdult, R.; Garcia, F.D.; Ege, B. Dismantling Megamos Crypto: Wirelessly Lockpicking a Vehicle Immobilizer. In Proceedings of the USENIX Security Symposium, Washington, DC, USA, 14–16 August 2013; pp. 703–718.
13. Woo, S.; Jo, H.J.; Lee, D.H. A Practical Wireless Attack on the Connected Car and Security Protocol for In-Vehicle CAN. *IEEE Trans. Intell. Transp. Syst.* **2014**, 1–14. [CrossRef]
14. Miller, C.; Valasek, C. Adventures in automotive networks and control units. *Def Con* **2013**, *21*, 260–264.
15. Miller, C.; Valasek, C. Remote exploitation of an unaltered passenger vehicle. *Black Hat USA* **2015**, *2015*, 91.
16. Koscher, K.; Czeskis, A.; Roesner, F.; Patel, S.; Kohno, T.; Checkoway, S.; McCoy, D.; Kantor, B.; Anderson, D.; Shacham, H.; et al. Experimental Security Analysis of a Modern Automobile. In Proceedings of the 2010 IEEE Symposium on Security and Privacy, Berkeley/Oakland, CA, USA, 16–19 May 2010; pp. 447–462.
17. Checkoway, S.; McCoy, D.; Kantor, B.; Anderson, D.; Shacham, H.; Savage, S.; Koscher, K.; Czeskis, A.; Roesner, F.; Kohno, T.; et al. Comprehensive Experimental Analyses of Automotive Attack Surfaces. In Proceedings of the USENIX Security Symposium, San Francisco, CA, USA, 8–9 August 2011.
18. SAE Vehicle Electrical System Security Committee. *SAE J3061-Cybersecurity Guidebook for Cyber-Physical Automotive Systems*; SAE International: Warrendale, PA, USA, 2016.
19. Barber, A. *Status of work in Process on ISO/SAE 21434 Automotive Cybersecurity Standard*; ISO SAE International: Geneva, Switzerland, 2018; Volume 10.
20. Upstream Security Ltd. Smart Mobility Cyber Attacks Repository, 2019. Available online: <https://www.upstream.auto/research/automotive-cybersecurity/> (accessed on 28 March 2019).
21. Hochschule Karlsruhe - Technik und Wirtschaft. Sichere Datenverarbeitung beim autonomen Fahren: Starke IT-Sicherheit für das Auto der Zukunft—Forschungsverbund entwickelt neue Ansätze. Available online: <https://www.hs-karlsruhe.de/presse/secforcars/> (accessed on 28 March 2019).
22. Bundesministerium für Bildung und Forschung. SecForCARS: Sicherheit für vernetzte, autonome Fahrzeuge. Available online: <https://www.forschung-it-sicherheit-kommunikationssysteme.de/projekte/sicherheit-fuer-vernetzte-autonome-fahrzeuge> (accessed on 28 March 2019).
23. Lee, E.A. Cyber Physical Systems: Design Challenges. In Proceedings of the IEEE Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing, Orlando, FL, USA, 5–7 May 2008; pp. 363–369.
24. King, J.D. Passive Remote Keyless Entry System. US Patent 6,236,333, 22 May 2001.
25. Thomas, E.; Timo, K.; Amir, M.; Christof, P.; Mahmoud, S.; Mohammad, T.M.S. On the Power of Power Analysis in the Real World: A Complete Break of the KeeLoq Code Hopping Scheme. In *CRYPTO 2008, Volume 5157 of LNCS*; LNCS, Ed.; Springer: Berlin, Germany, 2008; pp. 203–220.

26. Rüsberg, K. Keyless Gone: Autodiebe Tricksen Kontaktlose Schließsysteme aus, 2015. Available online: <https://www.heise.de/ct/ausgabe/2015-26-Autodiebe-tricksen-kontaktlose-Schliesssysteme-aus-3013915.html> (accessed on 28 March 2019).
27. Courtois, N.T.; Bard, G.V.; Wagner, D. Algebraic and Slide Attacks on KeeLoq. In *Fast Software Encryption*; Nyberg, K., Ed.; Springer: Berlin/Heidelberg, Germany, 2008; Volume 5086, pp. 97–115.
28. Miller, C.; Valasek, C. Can message injection. In *OG Dynamite Edition*; 2016. Available online: <http://illmatics.com/can%20message%20injection.pdf> (accessed on 17 April 2019).
29. Hoppe, T.; Kiltz, S.; Dittmann, J. Security Threats to Automotive CAN Networks—Practical Examples and Selected Short-Term Countermeasures. In *Computer Safety, Reliability, and Security*; Harrison, M.D., Suján, M.A., Eds.; Springer: Berlin, Germany; New York, NY, USA, 2008; Volume 5219, pp. 235–248.
30. Miller, C.; Valasek, C. A survey of remote automotive attack surfaces. *Black Hat USA* **2014**, 2014, 94.
31. Foster, I.D.; Prudhomme, A.; Koscher, K.; Savage, S. Fast and Vulnerable: A Story of Telematic Failures. In Proceedings of the Workshop on Offensive Technologies (WOOT) Washington, DC, USA, 10–11 August 2015.
32. Mahaffey, K. Hacking a Tesla Model S: What We Found and What We Learned, 2015. Available online: <https://blog.lookout.com/hacking-a-tesla> (accessed on 28 March 2019).
33. Spill, D.; Bittau, A. BlueSniff: Eve Meets Alice and Bluetooth. *WOOT* **2007**, 7, 1–10.
34. Committee, S.O.R.A.V.S. *Taxonomy and Definitions for Terms Related to on-Road Motor Vehicle Automated Driving Systems*; SAE International: Warrendale, PA, USA, 2014.
35. Sitawarin, C.; Bhagoji, A.N.; Mosenia, A.; Chiang, M.; Mittal, P. DARTS: Deceiving Autonomous Cars with Toxic Signs. Available online: <http://arxiv.org/pdf/1802.06430v3> (accessed on 28 March 2019).
36. Eykholt, K.; Evtimov, I.; Fernandes, E.; Li, B.; Rahmati, A.; Xiao, C.; Prakash, A.; Kohno, T.; Song, D. Robust Physical-World Attacks on Deep Learning Models. Available online: <http://arxiv.org/pdf/1707.08945v5> (accessed on 28 March 2019).
37. Sommer, F.; Dürrwang, J. IEEM-HsKA/AAD: Automotive Attack Database (AAD). 2019. Available online: <https://github.com/IEEM-HsKA/AAD> (accessed on 28 March 2019).
38. Booth, H.; Rike, D.; Witte, G. The National Vulnerability Database (NVD): Overview. Available online: <https://nvd.nist.gov/> (accessed on 28 March 2019).
39. Mitre, C. Common Vulnerabilities and Exposures, 2005. Available online: <https://cve.mitre.org/> (accessed on 28 March 2019).
40. Australia Cyber Emergency Response Team. Security Bulletins. Available online: <https://www.auscert.org.au/bulletins/> (accessed on 28 March 2019).
41. Computer Emergency Response Team Coordination Center of China. China National Vulnerability Database (CNVD). Available online: <http://www.cnvd.org.cn/> (accessed on 2 March 2019).
42. China Information Security Evaluation Center. China National Vulnerability Database of Information Security (CNNVD). Available online: <http://www.cnnvd.org.cn/> (accessed on 2 March 2019).
43. Packet Storm. Exploits. Available online: <https://packetstormsecurity.com/> (accessed on 28 March 2019).
44. Offensive Security. Exploit Database. Available online: <https://www.exploit-db.com/> (accessed on 28 March 2019).
45. Rapid7. Vulnerability and Exploit Database. Available online: <https://www.rapid7.com/db/> (accessed on 28 March 2019).
46. Smiraglia, R.P. *The Elements of Knowledge Organization*; Springer: Berlin, Germany, 2014.
47. Bundesamt für Sicherheit in der Informationstechnik. Leitfaden IT-Forensik. Available online: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Cyber-Sicherheit/Themen/Leitfaden_IT-Forensik.pdf?__blob=publicationFile&v=2 (accessed on 28 March 2019).
48. Weber, S.; Karger, P.A.; Paradkar, A. A software flaw taxonomy: Aiming tools at security. *ACM SIGSOFT Softw. Eng. Notes* **2005**, 30, 1–7. [CrossRef]
49. Abbott, R.P.; Chin, J.S.; Donnelly, J.E.; Konigsford, W.L.; Tokubo, S.; Webb, D.A. Security analysis and enhancements of computer operating systems. Available online: <https://nvlpubs.nist.gov/nistpubs/Legacy/IR/nbsir76-1041.pdf> (accessed on 17 April 2019).
50. Aslam, T. A Taxonomy of Security Faults in the Unix Operating System. Master’s Thesis, Purdue University, West Lafayette, Indiana, 1995.

51. Aslam, T.; Krsul, I.; Spafford, E.H. *Use of a Taxonomy of Security Faults*; 1996. Available online: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.140.5631> (accessed on 17 April 2019).
52. Bisbey, R.; Hollingworth, D. *Protection Analysis: Final Report*; ISI/SR-78-13, Information Sciences Inst; 1978; Volume 3. Available online: <https://www.ncjrs.gov/App/Publications/abstract.aspx?ID=57682> (accessed on 17 April 2019).
53. Landwehr, C.E.; Bull, A.R.; McDermott, J.P.; Choi, W.S. A taxonomy of computer program security flaws. *ACM Comput. Surv.* **1994**, *26*, 211–254. [[CrossRef](#)]
54. Novak, J.; Krajnc, A.; Žontar, R. Taxonomy of static code analysis tools. In Proceedings of the 33rd International Convention MIPRO, Opatija, Croatia, 24–28 May 2010; pp. 418–422.
55. Lindqvist, U.; Jonsson, E. How to systematically classify computer security intrusions. In Proceedings of the IEEE Symposium on Security and Privacy, Oakland, CA, USA, 4–7 May 1997; pp. 154–163.
56. Neumann, P.G.; Parker, D.B. A summary of computer misuse techniques. In Proceedings of the 12th National Computer Security Conference, Baltimore, MD, USA, 10–13 October 1989; pp. 396–407.
57. Howard, J.D. *An Analysis of Security Incidents on the Internet 1989–1995*; Carnegie-Mellon Univ.: Pittsburgh, PA, USA, 1997.
58. Howard, J.D.; Longstaff, T.A. *A Common Language for Computer Security Incidents*; Sandia National Labs.: Livermore, CA, USA, 1998.
59. Hoppe, T.; Dittman, J. Sniffing/Replay Attacks on CAN Buses: A simulated attack on the electric window lift classified using an adapted CERT taxonomy. In Proceedings of the 2nd Workshop on Embedded Systems Security (WESS), Salzburg, Austria, 4 October 2007; pp. 1–6.
60. Thing, V.L.L.; Wu, J. Autonomous vehicle security: A taxonomy of attacks and defences. In Proceedings of the 2016 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Chengdu, China, 15–18 December 2016; pp. 164–170.
61. Bundesamt für Sicherheit in der Informationstechnik. Cyber-Sicherheit: BSI—Leitfaden IT-Forensik. Available online: https://www.bsi.bund.de/DE/Themen/Cyber-Sicherheit/Dienstleistungen/IT-Forensik/forensik_node.html (accessed on 28 March 2019).
62. Arkin, B.; Stender, S.; McGraw, G. Software penetration testing. *IEEE Secur. Priv.* **2005**, *3*, 84–87. [[CrossRef](#)]
63. Kohnfelder, L.; Garg, P. The STRIDE Threat Model. Available online: [https://docs.microsoft.com/en-us/previous-versions/commerce-server/ee823878\(v=cs.20\)](https://docs.microsoft.com/en-us/previous-versions/commerce-server/ee823878(v=cs.20)) (accessed on 28 March 2019).
64. ISO 14229: 2006(E)—Road Vehicles—Unified Diagnostic Services (UDS)—Specification and Requirements. 2015. Available online: <https://www.iso.org/standard/45293.html> (accessed on 17 April 2019).
65. ISO 11898-1: 2015—Road Vehicles—Controller Area Network (CAN)—Part 1: Data Link Layer and Physical Signalling; 2015. Available online: <https://www.iso.org/standard/63648.html> (accessed on 17 April 2019).
66. Katz, J.; Menezes, A.J.; van Oorschot, P.C.; Vanstone, S.A. *Handbook of Applied Cryptography*; CRC Press: Boca Raton, FL, USA, 1996.
67. Charniak, E. *Introduction to Artificial Intelligence*; Pearson Education: Delhi, India 1985.
68. Huang, S.; Papernot, N.; Goodfellow, I.; Duan, Y.; Abbeel, P. Adversarial Attacks on Neural Network Policies. Available online: <http://arxiv.org/pdf/1702.02284v1> (accessed on 28 March 2019).
69. Shostack, A. STRIDE chart, 2007. Available online: <https://www.microsoft.com/security/blog/2007/09/11/stride-chart/> (accessed on 28 March 2019).
70. CVSS Special Interest Group. Common Vulnerability Scoring System v3.0: Specification Document, 2019. Available online: <https://www.first.org/cvss/specification-document> (accessed on 28 March 2019).
71. Tsipenyuk, K.; Chess, B.; McGraw, G. Seven pernicious kingdoms: A taxonomy of software security errors. *IEEE Secur. Priv.* **2005**, *3*, 81–84. [[CrossRef](#)]
72. Christey, S. *PLOVER- Preliminary List Of Vulnerability Examples for Researchers*; 2006. Available online: <https://cwe.mitre.org/documents/sources/PLOVER.pdf> (accessed on 28 March 2019).
73. MITRE, C.W. Common weakness enumeration, 2006. Available online: <https://cwe.mitre.org/> (accessed on 28 March 2019).
74. Zimmermann, W.; Schmidgall, R. *Bussysteme in der Fahrzeugtechnik: Protokolle, Standards und Softwarearchitektur*; Springer Vieweg: Wiesbaden, Germany, 2014.
75. Popovici, K.; Jerraya, A. Hardware Abstraction Layer. In *Hardware-Dependent Software*; Springer: Berlin, Germany, 2009; pp. 67–94.

76. Winter, M.; Goetz, H.; Brandes, C.; Roßner, T. *Basiswissen Modellbasierter Test*; dpunkt.verlag GmbH: Heidelberg, Germany, 2012.
77. Ruddle, A.; Ward, D.; Weyl, B.; Idrees, S.; Roudier, Y.; Friedewald, M.; Leimbach, T.; Fuchs, A.; Grgens, S.; Henniger, O.; et al. Deliverable d2. 3: Security Requirements for Automotive On-Board Networks Based on Dark-Side Scenarios. 2009. Available online: <https://evita-project.org/deliverables.html> (accessed on 28 March 2019).
78. Kerckhoffs, A. La cryptographie militaire. *J. Sci. Mil.* **1883**, 9, 38.
79. Offensive Security. Kali Linux Tools Listing. 2019. Available online: <https://tools.kali.org/tools-listing> (accessed on 28 March 2019).
80. Kennedy, D.; O’gorman, J.; Kearns, D.; Aharoni, M. *Metasploit: The Penetration Tester’s Guide*; No Starch Press: San Francisco, CA, USA, 2011.
81. Center, C.C. CERT Vulnerability Notes Database. 2001. Available online: <https://www.kb.cert.org/vuls/> (accessed on 28 March 2019).
82. ISO 26262. Road Vehicles—Functional Safety. 2018. Available online: <https://www.iso.org/standard/68383.html> (accessed on 28 March 2019).
83. Ring, M.; Dürrwang, J. Angriffsklassifizierung. Available online: www.mmt.hs-karlsruhe.de/downloads/IEEM/Angriffsklassifizierung.ods (accessed on 28 March 2019).
84. Bishop, M.; Bailey, D. *A Critical Analysis of Vulnerability Taxonomies*; 1996. Available online: <http://www.seifsec.com/unclassified/hacking/A%20Critical%20Analysis%20of%20Vulnerability%20Taxonomies.pdf> (accessed on 17 April 2019).



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).