information

MDPI

Article

SAVTA: A Hybrid Vehicular Threat Model: Overview and Case Study

Mohammad Hamad [1,*] and Vassilis Prevelakis [2]

[1] Department of Electrical and Computer Engineering, Technical University of Munich, 80333 Munich, Germany
[2] Institute of Computer and Network Engineering, Technical University of Braunschweig, 38106 Braunschweig, Germany; prevelakis@ida.ing.tu-bs.de
* Correspondence: mohammad.hamad@tum.de

Received: 11 March 2020; Accepted: 15 May 2020; Published: 19 May 2020

Abstract: In recent years, significant developments were introduced within the vehicular domain, evolving the vehicles to become a network of many embedded systems which depend on a set of sensors to interact with each other and with the surrounding environment. While these improvements have increased the safety and incontestability of the automotive system, they have opened the door for new potential security threats which need to be defined, assessed, and mitigated. The SAE J3061 standard has defined threat modeling as a critical step toward the secure development process for vehicle systems, but it did not determine which method could be used to achieve this process. Therefore, many threat modeling approaches were adopted. However, using one individual approach will not identify all the threats which could target the system, and may lead to insufficient mitigation mechanisms. Thus, having complete security requires the usage of a comprehensive threat model which identifies all the potential threats and vulnerabilities. In this work, we tried to revise the existing threat modeling efforts in the vehicular domain. Also, we proposed using a hybrid method called the Software, Asset, Vulnerability, Threat, and Attacker (SAVTA)-centric method to support security analysis for vehicular systems. SAVTA combines different existing threat modeling approaches to create a comprehensive and hybridized threat model. The model is used as an aid to construct general attack trees which illustrate attack vectors that threaten a particular vehicle asset and classify these attacks under different sub-trees.

Keywords: threat modeling; automotive security

1. Introduction

In recent years, vehicles manufacturing has changed significantly: vehicles moved from a largely electro-mechanical system into an Electrical and Electronic (E/E) system. This can be seen in the increased use of automotive embedded systems and the large quantity of embedded software which is integrated within every single vehicle [1,2]. A modern car contains 100–70 micro controller-based computers known as Electronic Control Unit (ECU). Each ECU relies on a set of sensors and actuators to serve one or more of the E/E systems or subsystems in the vehicle. These ECUs are interconnected using different bus systems such as Controller Area Network (CAN), FlexRay, and Ethernet [3,4]. In addition, modern vehicles are equipped with various technologies, such as WiFi, 5G, Global Positioning System (GPS), and Bluetooth, giving them the capability to collaborate and communicate with each other and with roadside units.

The increase of the vehicle connectivity is a double-edged sword. On the one hand, it extends the vehicle's functionalities and capabilities, but on the other hand, it opens the door to several cybersecurity threats and makes the vehicle a more attractive target for adversaries. Almost all

vehicle vendors have suffered from security weaknesses within their produced vehicles. For example, there were vulnerabilities in Chrysler's Uconnect software [5], Skoda's SmartGate system [6], BMW's ConnectedDrive [7], the WIFI access point of the Mitsubishi Outlander plugin hybrid electric vehicle (PHEV) [8], GM's Onstar [9], and many others. Those vulnerabilities gave attackers the chance to perform numerous attacks and many malicious actions such as turning on/off air conditioning, heating, and lights, disabling the theft alarm and so forth. They also caused the recall of millions of cars.

Therefore, the need to develop systematic mechanisms to ensure the security of the vehicle components, similar to those used to ensure safety, becomes critical. Developing such mechanisms requires determining the security requirements, vulnerabilities and threats which the system faces, and the attackers who might target it. Threat modeling helps to identify and address most of the potential threats. In fact, threat identification would likely reduce the life cycle as well as the cost of achieving security objectives when it is considered during the design process. Furthermore, threat modeling provides relevant information about the attack vectors which threaten the system. Such information can be used as a reference during the test process to avoid omitted threats. The SAE J3061 standard (Cybersecurity Guidebook for Cyber–Physical Automotive Systems) [10] has determined the threat analysis as an important step during the design and development of cybersecurity aware automotive system.

Although SAE J3061 standard emphasized the need for the threat analysis process, it did not determine which method could be used to achieve this process. The SAE J3061 standard leaves to an organization to determine which threat analysis method is appropriate for its purposes. As a result of that, many threat modeling methods were adopted. However, most of the research examines the potential threats only partially by focusing on a certain aspect or by looking at threats which affect a particular sub-system independently. Practically speaking, the lack of a general threat model within the vehicular domain makes threat analysis for the different subsystems a resource-consuming task. Additionally, it increases the possibility of inconsistencies between the interacting subsystems and causes redundancy when defining the attack vectors.

In this work, we revise the various existing threat modeling approaches and their usage in the vehicular domain. We use these approaches to develop a hybridized threat model for the automotive domain. Our model seeks to combine multiple approaches to arrive at a more comprehensive one. Within our model, we start looking at the potential attacker agents by defining various groups of attackers and their motivations. Then, we identify the potential targets that they may threaten and the security requirements which are required to protect these targets. Also, we propose an abstract model that can be used to classify all conceivable attacks against the vehicular domain. The abstract model is used as an aid to constructing attack trees [11] which illustrate attack vectors that threaten a particular vehicle asset and classify them under different sub-trees. The main advantage of the threats compartmentalization is the ability to conclude the effective defense mechanisms against these threats.

The rest of the paper is organized as follows: In Section 2, we review existing threat models in many IT domains including those which were proposed to be used with the vehicular domain. Then, we present our proposed threat model in Section 3 which explains the various SAVTA components and presents an abstract model based on SAVTA to identify possible threats within the automotive system. In Section 4, we discuss some points which need to be considered during the automotive risk analysis using the generated attack tree. In Section 5, we use our abstract model to identify some threats within the autonomous vehicle driving use case. Finally, we present our conclusions in Section 6.

2. Foundations

2.1. Terminology

Before we can explain what a threat model is, it is necessary to define some basic definitions that we will need based on [12,13]. The first definition is the *asset* which represents any system resource that needs or intends to be protected against an adversary. The asset could include software, hardware, data, and so forth. Any flaw or weakness in asset or system's design, implementation, or operation and management that could be exploited by the attacker and violate the system's security policy, is called *vulnerability*. A *threat* is the potential cause of an unwanted incident, which can result in harm to a system. An *attack* against one asset is any malicious activity that attempts to collect, expose, alter, disrupt, disable, degrade, destroy or gain unauthorized access to this asset by exploiting one or more vulnerabilities.

Based on the previous definitions, we next provide the definition of threat modeling:

Definition 1. *Threat Modeling: A systematic process used to (a) analyze the potential attackers, (b) identify the security vulnerabilities and threats in the (sub-)system assets which could be exploited by an attacker and (c) provide significant information that would help to safeguard the target (sub-)system against attacks as well as to develop realistic and meaningful security requirements for this (sub-)system.*

2.2. Threat Modeling Approaches

Different approaches were used to develop threat modeling. Each of these approaches focuses on a certain aspect of the system such as asset, attacker, software, or system vulnerabilities [14,15]. Typically, threat modeling has been implemented using one of these different approaches independently. Next, we will look at each of these approaches in turn and outline how they have been adopted in the automotive domain.

2.2.1. Attacker-Centric

This approach focuses on profiling attackers' characteristics, goals, skills, and motivations for targeting the studied system. By putting ourselves in the shoes of the system attackers, we can understand their goals; this will provide us with valuable information about the most targeted assets in our system. Consequently, this gives us a better understanding to implement appropriate mitigation strategies for stopping these attackers.

Intel Threat Agent Library (TAL) [16] is one example of attacker-centric threat models. TAL standardized a list of 22 threat agents that pose threats to IT systems. The classification of the agent is based on eight unique attributes: the intent of the agent whether she intends to cause harm to the system or not, the way the agent can access the assets, the agent's main outcomes, the limits which constrain the agent, the resources available to that agent to carry out the attack, the skills that the attacker must have to target the system, the agent's objective, and the identity visibility of the threat agent. In addition, for each of these agents, the model keeps an updated rating based on many factors such as that agent's recent attacking activities.

Intel's Threat Agent Risk Assessment (TARA) [17] is one of the main examples of threat modeling methodologies that consider the threat agent as the origin of the security risk. TARA aims to narrow the field of all possible attacks against the system and determine the most probable attacks based on the TAL agent list and the rate which is given for each agent.

Within our previous work [18], we were among the first authors to attempt to concentrate on the attacker-centric method for the automotive domain by classifying attackers against the vehicular system and trying to define their different goals. Other efforts also followed the same strategy and included the attacker in their threat model, such as [19,20]. One of the most interesting efforts was by Karahasanovic et al. [21]; the authors proposed the adaptation of the TARA threat model for

the automotive system. To achieve that, they made some changes to TARA to make it fit with the automotive industry's needs.

2.2.2. Assets-Centric

This approach focuses on system assets to protect them from different attackers. Assets can refer to anything valuable in the system. The idea behind this approach is that by profiling assets that are more attractive to attackers and identifying their vulnerabilities, the organization can wisely invest in protecting these assets and does not waste resources guarding noncritical or unattractive assets.

Operationally Critical Threat, Asset, and Vulnerability Evaluation (OCTAVE) [22] is an example of asset-centric threat modeling methodology. This approach consists of four phases: during the first phase, the organization develops risk measurement criteria. In the second phase, the critical assets are profiled and their security requirements are defined. Identifying the threats to these assets is accomplished within the third phase. Finally, analyzing the risks related to the different assets and starting the development of mitigation approaches take place during the fourth phase.

The Threat, Vulnerability, and Risk Analysis (TVRA) model, which was proposed by the European Telecommunications Standards Institute (ETSI), aims to support the security analysis in the vehicular domain by focusing on the assets of the system [23]. The TVRA process starts by defining the security objectives and requirements of the system. Then, it defines a list containing all the system's assets. For each asset in the defined list, classification of all possible vulnerabilities and related threats is carried out. The risks of threats are determined based on the likelihood of the identified threats. Finally, countermeasures are proposed to reduce the risks of the threats.

2.2.3. Vulnerability and Threat-Centric

Instead of considering all existing vulnerabilities in different assets, this approach aims to focus on the risky vulnerabilities which are most reachable and desirable for attackers and to develop mitigation for them. Usually, such vulnerabilities exist within the assets that attackers can easily reach (i.e., attack surfaces or attack entry points).

Skybox Threat–Centric Vulnerability Management (TCVM) [24] is an example of such an approach. The TCVM process starts by profiling all existing vulnerabilities in the system's assets. The next step is identifying the exposed and exploitable vulnerabilities from the existing ones based on context-based prioritization and management techniques.

In the vehicular domain, many researchers have focused on identifying the vulnerabilities of the various vehicular system assets. Checkoway et al. [25] have studied potential attack surfaces of the vehicle which could be exploited by attackers externally. On the other hand, Koscher et al. [26] investigated the attack surfaces on the underlying system structure. Both works demonstrated that attackers could leverage direct access to the CAN bus to control various functions by exploiting vulnerabilities in the attack surface.

2.2.4. Software-Centric

This is also called design-centric. It can be considered as a special case of Assets-centric method since it focuses on a certain type of assets (i.e., the software component). As the name suggests, this approach focuses on security during the design phase of software components for the system by identifying the threats which may expose each component. Loren Kohnfelder and Praerit Garg at Microsoft proposed a software-centric methodology for threat modeling called STRIDE [27], which stands for the major six attack categories which threaten software products: spoofing of user identity, tampering with data, repudiation, information disclosure, denial of service and elevation of privilege.

Winsen [28] proposed using STRIDE to identify all possible threats for future autonomous and connected vehicles. Then, he analyzed these threats based on their severity and controllability. Macher et al. [29] proposed combining STRIDE with safety analysis (hazard analysis and risk

assessment (HARA)) to ensure a security-aware safety development for automotive system. In the same direction, Monteuuis et al. have extended STRIDE to develop a security analysis framework called SARA [30]. NCC Group has proposed a template for automotive threat modeling [31], which could be used through Microsoft Threat Modeling Tool [32]. The proposed template classifies the threats based on the STRIDE approach. The template tries to prioritize the threat based on the risk of every presented threat. Besides, it includes a possibility to suggest some mitigation mechanisms against each threat. Ma et al. [33] have also adopted STRIDE approach to create a practical and efficient automotive threat modeling by extending the NCC group template. HEAVENS (Healing Vulnerabilities to Enhance Software Security and Safety) project has proposed a security model based on STRIDE approach [34].

2.2.5. Attack Trees

Threat analysis describes who the potential attacker is, what are the motivations behind an attack are, and which components could be threatened. Describing how an attack could be executed is the mission of attack trees [11]. An attack tree is used to explain attacks in a tree structure as shown in Figure 1. The root of the tree represents the attacker's ultimate goal, while the intermediate nodes of the tree (sub-goals) define different stages of the attack. In the case that a node in an attack tree requires all of its sub-goals to be achieved, the sub-goals are combined by an AND branch. If a node requires that any of its sub-goals be achieved, the sub-goals are combined by an OR branch. Leaf nodes represent atomic attacks. Attack scenarios are generated from the attack tree by traversing the tree in a depth-first method [35]. Each attack scenario will contain the minimum combination of leaves. In classical attack tree models the attack chronology is disregarded. However, in many cases, the success of an attack depends on the subsequent success of interrelated attack steps. In some circumstances, the specific order of the sub-attacks is vital to achieving the parent goal. To handle such special cases, a new sequential AND gate (SAND) was introduced by Arnold et al. [36]. This gate was influenced by the Priority-AND (PAND) gate, which was adopted in the fault trees domain a long time ago [37]. We will use the PAND gate in this paper whenever we need to show the order of attacks in our attack trees (Figure 1 shows two sub-attacks (AT4 and AT5) which are used as inputs for PAND gate).

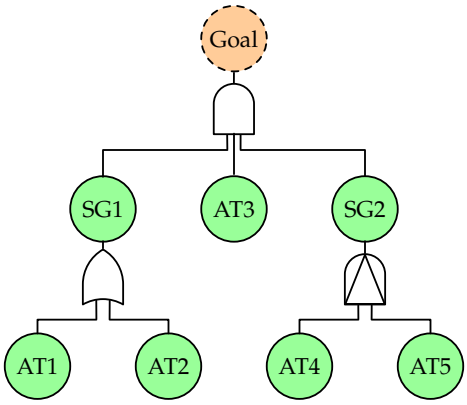

Figure 1. The Attack tree structure.

Attack trees have been used as a tool to illustrate the attack steps for individual attack scenarios, to assess the risk of these attacks which could target the modern vehicle [18,38–40], or to support the planning of intrusion response system within the vehicular systems [41]. E-Safety Vehicle Intrusion Protected Applications (EVITA) proposed a method called threat and operability analysis (THROP). This method proposed the use of attack trees to identify attacks and formalize the security requirements for the automotive's on-board networks [42,43]. Aijaz et al. [44] tried to create reusable attack trees for V2V communication threats. In our work, we are attempting to provide an abstract model which helps to create general attack trees for the entire vehicular domain.

3. SAVTA

As we showed in the previous section, there are many threat modeling approaches that have been implemented within the vehicular domain. Using one individual approach will not identify all the threats which could target the system and may lead to insufficient mitigation mechanisms [45,46]. For example, applying an asset-centric method requires the definition of the most critical assets that an attacker may target. The ignorance of the type of attacker (whether she is internal such as the driver, or external) and its capabilities will affect the selection of the potential attack surfaces from these assets.

Effective defense against threats requires addressing all existing security flaws in the target system and identifying threats that exploit these vulnerabilities. In addition, it demands a good comprehension of the prospective attackers, their capabilities, and their objectives. To address all of this, we propose a threat model called Software, Asset, Vulnerability and Threat, and Attacker (SAVTA)- centric method) that combines different existing approaches, bringing them together to create a comprehensive threat model for the vehicular systems. We call this a hybrid threat model.

Figure 2 shows a general view of SAVTA threat model and how the various approaches are interconnected with each other. This interaction reflects the threat model definition (i.e., Definition 1) which was given in the previous section. Within a very complex environment such as a vehicle, different assets coexist. These assets usually suffer from several hidden vulnerabilities. A motivated attacker could target each of these assets by generating suitable conditions to exploit one or more of these vulnerabilities. Exploiting any of these vulnerabilities always ends up with a violation of one or more of the security requirements.

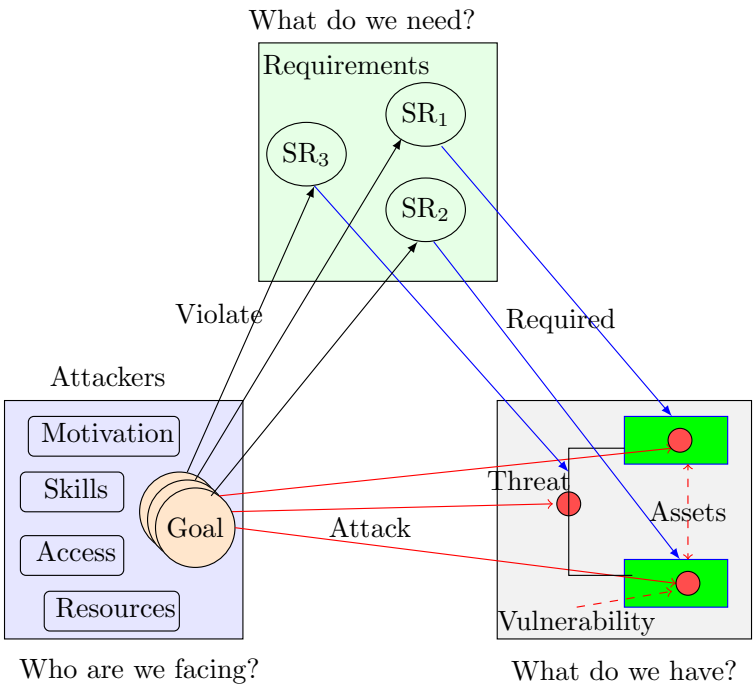

Figure 2. Software, Asset, Vulnerability, Threat, and Attacker (SAVTA) threat model.

The next steps are required during application of the SAVTA threat model (Figure 2):

- We start the process by looking at what are we facing? We can address this question by defining the Attacker Profile which includes information about all possible attackers who may target the vehicular system. We classify these attackers based on their motivation and the other resources that they have or use to reach their goal.
- In addition, we have to answer the next main question what do we have? The answer to this question should cover all the possible weak points that we have in our system. We achieve that

by identifying all the vehicle's attackable assets; and for each of them, we have to identify all the vulnerabilities and threats which affect it. We also need to define the relations which interconnect the different assets. These relations are a reflection of the functional requirements that the system needs to work. By defining the relations between the different assets, we can point out the assets on which all others depend for accomplishing the objective (i.e., the centre of gravity [47,48]) as well as the possible attack chain which leads to defining all attack surfaces.

- The last question we need to answer is what do we need to secure these assets? Security requirements need to be determined for each asset as well as the relationship between them. Determining these requirements involves defining the references for the security mitigation implementation against attacks that may threaten the assets, for example, the integrity requirement of the communication link which could be threatened by a replay attack.
- After obtaining the previous information, we start linking them together to create an abstract model which can be used to create attack trees which will be adopted to classify and identify all the vulnerabilities, threats, and attacks. It is important to note that the three previous steps can be achieved in parallel and carried out by different security teams.

Next, we discuss each of these steps in detail.

3.1. Attacker Profile

Different groups of attackers are attracted to attacking vehicles. These groups range from the owner of the car to an expert hacker with sophisticated tools. One of the most important steps towards securing the vehicular system is knowing the goal of threat agents (i.e., attackers) as well as other properties of each agent, such as motivation, skills, resources, and the accessibility of the attacked object, which can play a significant role in reaching this goal. Understanding these properties for each attacker will enrich our knowledge and improve our capabilities to define adequate mitigation. We start defining the possible threat agents for the automotive domain by looking at the motivations behind the various attacks that we have faced:

- Falsification: An attacker (who could be the owner or the driver) may wish to misrepresent actual vehicle information such as changing the achograph or odograph measurements to sell the car with a false mileage reading or to defraud the operator and safety regulations [49].
- Illegal profit: An attacker could make a profit by stealing the vehicle or by selling the attack capability to a different organization. Some attacks could be driven by a commercial competitor of the target vehicle's vendor to sabotage their product and gain market share. Although there is no published case which states that a particular attack has been carried out based on industrial espionage, such motivation remains valid since there is a history of industrial espionage between vehicle manufacturers [50]. Such an attack requires the collusion of one or more insiders in the targeted organization.
- Fun and vandalism: Revenge and vandalism could motivate some attacks, as in the case of a dismissed employee who sought to punish his ex-company by bricking cars sold by this company [51].
- Research and test purposes: Attacks and penetration tests could be performed by security experts or test teams. The attackers, in this case, have benign motivations. They seek to discover security flaws in different components of the vehicle systems before they can be exploited by third parties.
- Terrorist: There are still no real incidents in which such motivation has been proven to be behind the vehicle cyberattack; Richard Clarke, a former U.S. National Coordinator for Security Infrastructure Protection and Counterterrorism, hinted that the fatal crash of journalist Michael Hastings' Mercedes C250 coupé is consistent with a car cyberattack [52].
- Overlap: Sometimes, multiple motives could lie behind a single attack.

It is important to note that this list is not complete; other motivations can be added whenever we discover a new attack. However, motivation alone is not enough. An attacker needs sufficient

resources to achieve his goals. These resources include: skills, capabilities, technical equipment, and financial resources. The disparity in these resources could be used as a tool for adding another level of classification of the attackers:

- Limited: Attackers in this group have limited financial resources and insignificant knowledge about the vehicle architecture. Such attackers lack the ability to use complicated tools. Regular car thieves, owners who would like to install or replace a component within their cars, an attacker who tampers with highway signals to gain a reputation in their community, unsophisticated attackers (script kiddie) and others are good examples of members of this group.
- Adequate: This group includes highly skilled experts who have adequate tools and equipment to perform the attack. However, they may not have sufficient financial resources to support them or to build bigger teams and cause huge damage. However, the members of this group could use their experience to obtain profit, such as by operating as black-hat hackers. Mechanics, owners with good knowledge and security researchers belong to this group.
- Sophisticated: This group contains expert hackers with sophisticated tools and huge financial support. Good examples of this group are the cybersecurity organizations (governmental and nongovernmental) who have multiple members of the above group who work together. Typically, massive financial support enables them to obtain sophisticated tools and attract experts. Some security research groups with similar resources could be another example of this class.

3.2. Attackable Assets

Attackers may focus on different parts of the vehicle components. One of the main steps of the SAVTA method is to determine these assets and identify the vulnerabilities and attacks which threaten them. In autonomous vehicles, a software component (SWC) in one ECU depends on the information provided by a set of sensors or transmitted by other components in another ECU, to perform its function. This function could be translated into a physical action by different actuators or transmitted as an input for another component in another ECU. For such a system and based on the definition of the asset, we can identify the next vehicle assets (see Figure 3) which may be targeted by the different threat agents and that need to be protected:

- In-Vehicle Hardware: ECUs represent the main hardware target within the modern vehicle. One primary attack against the ECU is a side channel attack. By carrying out such attacks, the attacker can gather information during the execution of the crypto-system and use of the information gathered to extract secure critical information such as crypto keys [53–55]. In addition, reaching the in-vehicle hardware gives the attacker the opportunity to replace any legitimate device with a malicious one, or even to install new hardware which could cause havoc.

  ECUs are not the only hardware components that could be targeted by attackers. Other devices that are connected to the vehicle, such as phones, tablets, diagnostic devices connected via ODB-II, etc., can be used as a gateway to attack a vehicle if it contains a security vulnerability [38]. Woo et al. [56] demonstrated the possibility of attacking a vehicle remotely by using a malicious application installed on a smartphone connected to the victim's vehicle.

  The smart sensors (e.g., camera, LiDAR, radar, etc.) within the vehicle can be other hardware targets for various attacks such as camera sensor blinding attacks [57], LiDAR blinding attacks [58], and others. These attacks aim to prevent these sensors from working properly, which has a serious safety impact on both autonomous and normal vehicles.
- Software and Firmware: The massive amount of integrated software in each vehicle and the different levels of security auditing between the different vendors make the software more susceptible to attacks. Attackers can benefit from software vulnerabilities to inject malicious code, causing software components to behave maliciously or to stop the application and prevent the vehicle from achieving certain functionalities. The firmware which controls the ECU could be a

target for various attacks; some attackers could tamper with the ECU firmware to achieve superior performance [59].

Not only automotive applications but firmware itself can contain many vulnerabilities or unnecessary embodied services which could be exploited by an attacker [60,61]. Manipulating the firmware of ECU usually requires a direct access to the target ECU. ECUs are programmable devices that include some ports and serial consoles to help the developer access and maintain the firmware and software mapped to each ECU. The same ports and consoles can give attackers the ability to reflash the ECU with malicious custom firmware [62]. In 2015, Charlie Miller and Chris Valasek [5] were able to flash one of Cherokee Jeep's head unit chips with modified firmware (Chip tuning attack [63]). Later, they used the malicious firmware to send commands through the in-vehicle network to perform malicious actions such as disabling brakes, taking control of the steering wheel, and even stopping the engine.

Over the Air (OTA) firmware and software updates represent a new challenge as well as a predictable resource for introducing new vulnerabilities to the automotive system if not handled correctly. One malicious or wrong update can end up as a huge issue, as in the case of the 2016 Toyota Land Cruiser Enform system, which was continuously rebooting itself because of a new update containing errant data [64].

- Data: The huge amount of data exchanged, the architecture of the in-vehicle network used to transfer this data, and the unsophisticated hardware (i.e., ECU) used to store it make data a very attractive asset for security attacks. Attackers can target data stored in some ECUs; this data could be the execution code of the applications, the firmware drivers, crypto-private keys, digital certificates, or private vehicle and driver activities (e.g., vehicle location, navigation destination, etc.). Extracting such data may occur by targeting the hardware through side-channel attacks, or by targeting one malicious software component or the Operating System (OS) itself, especially if that OS does not store the data properly with strict access restrictions. Alternatively, attackers could threaten *transmitted wired/wireless data* within the vehicle.

  (a) In-vehicle data exchanged between different components or between one component and its sensors or actuators. Attacking this data depends on the existence of vulnerabilities in the internal network protocols. Spoofing, altering, or drooping the transferred data between the on-board system and different sensors or actuators are examples of attacks against such data [65].
  (b) Data transferred between the vehicle and the external world using Vehicle-to-everything (V2X) communication technologies. This includes exchanged data between vehicles using Vehicle-to-Vehicle (V2V) communication or data transferred between the vehicle and surrounding infrastructure using Vehicle-to-Infrastructure (V2I) communication. This data could be targeted after it is received by the sensors; in this case, it is treated as in-vehicle exchanged data. In other cases, the data can be infected by different attacks (such as spoofing and emitting false data) before it has been revived by sensors [57]. Finally, the data could be targeted before it leaves its source.

- Surrounding infrastructure: Another critical asset that could be targeted by attacks is the surrounding environment of the vehicle. Although the surrounding environment, such as other vehicles, roadside units (RSUs), etc., is considered external to the in-vehicle system, it has a direct impact on the security of the in-vehicle system since it is the data source for sensors and V2X. Note that the under-study in-vehicle systems (or the vehicle as a unit) are considered external for other vehicles moving on the same road or for road infrastructure. Thus, the surrounding environment can be studied similarly to in-vehicle systems by looking at its threat agents as well as its assets (i.e., hardware, software and framework, data). Many security attacks can be launched against the surrounding infrastructure. A typical example of such an attack is adding stickers to traffic

signs [66]. Another example is modifications to electronic road signs, such as "Zombies Ahead", where an attacker figured out how to alter the text on electronic road signs to create warnings of a zombie attack. Even such a ridiculous attack could create public safety issues for drivers on the roadway [67].

We need to define a list of all these assets for each subsystem within the vehicle. The definition of these assets depends on the security analysis and how it sees the system as well as how much information it has about the different subsystems. Assets can be defined in a very abstract way by looking at each functionality of the system as a black box which will be mapped in a hardware platform and will intercommunicate with other functionalities using very generic network architecture. From such a viewpoint, the transmitted data will reflect the functional relationship between the different functionalities. A detailed overview can be achieved if the information about the software components as well as the actual network architecture is available. This detailed information can be derived from the abstract view by decomposing each defined functionality into its actual software components, run-time environment components, device drivers, protocols, etc. Another way could be by looking at these assets based on the different sub-domains (i.e., entertainment domain, body domain, powertrain domain, etc.) that they belong to. Many tools can be used to visually represent the different assets in the system in the different ways; one example of such tools could be data flow diagrams (DFDs) [15] (pp. 44–47). Any other equivalent type of diagram can also be used. Figure 3 shows one representation of the different system assets.

The main point here is to distinguish between the local (e.g., functionalities, sensors, actuators inside the vehicle) and external domains (e.g., nearby vehicles, RSUs, etc.). Functionalities (or components) which have relationships with external domains are usually more attractive targets and may represent reachable attack entry points for attackers because they are more accessible than others.

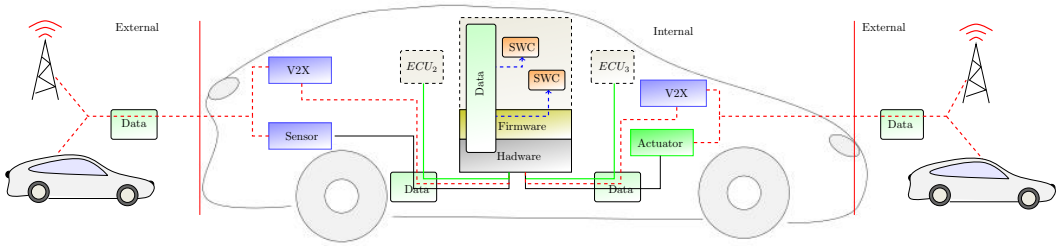

Figure 3. The different assets of the automotive system.

3.3. Attack Effects

The relationship between the different assets plays the main role in determining the effect of the attack on a specific component. For this contribution, we classify attacks based on their effect:

- Limited attack: The ultimate target of some attacks could be a single part of the vehicle. The effect of such attacks will remain limited to the attacked ECU(s) and not propagate any further. The targeted system will define the jeopardy level of the attack.
- Stepping stone attack: The attack can start by compromising one component or subsystem. Later, the attacker uses this subsystem as an attack surface to plague all related subsystems. The same process can be repeated for the newly infected components. Koscher et al. [26] showed that an attacker who can control one ECU is able to attack other connected ECUs.

3.4. Security Requirements

For each of the defined assets, a set of security requirements needs to be checked. These requirements are a reflection of the various threats which may threaten that asset. Thus, we check whether each asset suffers from one or more of the STRIDE model components. As we can see in Table 1, each of the STRIDE components violates one of the security attributes. CIA^3 (an extension of

CIA with Authorization and Authenticity) attributes are used as a source to determine the security requirements for each asset. Next we explain the security attributes of our extended model:

- Integrity: providing integrity within vehicular systems is composed of:
  - Providing hardware integrity to prevent and detect any hardware component fraud.
  - Providing framework and software integrity to ensure that only trusted code can be run and to prevent infected code and malware from running.
  - Providing data integrity to safeguard against any modifications to data during a transaction.
- Authentication: is used to ensure and verify the authenticity of an entity. This includes the identity of that entity as well as other properties such as its location. Another important property that needs to be authenticated is the freshness of the information. The receiver needs to be able to verify that the data has been created/sent by the sender at a claimed time.
- Authorization: determines whether a certain component is allowed to access or communicate a certain resource (i.e., who should talk to whom). Approving the request of that component depends on the authentication of the requester as well as the access control rules for the requested resource.
- Confidentiality and Privacy: while providing authentication for the exchanged messages in the vehicular domain is vital, providing confidentiality is often less important. For example, there is no critical reason to encrypt all the messages exchanged between the different ECUs inside the vehicle. Enforcing confidentiality for the exchanged data should not be mainly to prevent vehicle identification detection. The ability to identify the vehicle is feasible already by different mechanisms without the need to snoop on exchanged messages (such as identifying the vehicle by color, number plate, etc.) The primary goals should be preventing a leak of the driver's critical data (such as driver behavior, previous location) as well as guaranteeing that any observer is unable to efficiently link different messages coming from the same source. In addition, it is critical to ensure the privacy of the data collected by the different sensors (e.g, images captured by the camera). In some scenarios, confidentiality is required; for example, leaving valuable stored information (e.g., private keys) without any confidentiality protection may leave the entire vehicle security at stake if an attacker is able to extract this data.
- Availability: refers to the fact that the assets must be available even if the system is under attack. Availability is required particularly for safety-related applications which are integrated into the vehicle. Losing the availability of such applications may have serious consequences, and even threaten the lives of passengers. This property is a common concern of both safety and security, but they address it differently. While safety handles unintentional events which may lead to loss of availability, security focuses on intentional attacks.

Table 1. Mapping STRIDE with CIA^3 security attributes.

STRIDE	Security Attribute (CIA^3)	Explanation
Spoofing Identity	Authenticity	pretend to be someone else
Tampering with Data	Integrity	improper assets alterations
Repudiation	Non-Repudiation	trackless
Information Disclosure	Confidentiality	to access to confidential data
Denial of Service	Availability	disable or delay accessing an asset
Elevation of Privilege	Authorization	perform unauthorized actions

3.5. Attack Accessibility

The way in which threat agents can exploit a vulnerability in any one of the aforementioned assets is very important for determining the applicability of the attack and proposing the right mitigation against it. We have specified the following three possible cases:

- Direct access: Some attacks require direct (physical) access by the threat agent to the target vehicle. Replacing the hardware component or connecting a malicious third party to the in-vehicle network is an example of such attacks. Such direct access could be achieved while a vehicle is parked. In such circumstances, the attackers cannot access the in-vehicle system, but may still be able, for example, to attach an external device to the vehicle such as a GPS device to track the vehicle later, or to target the vehicle's immobilizer and electronic locks [68]. In some cases, taking the car to the service station for a regular check could become an avenue for direct access by attackers. In such cases, an attacker has full access to the in-vehicle system and could take advantage of existing physical interfaces (e.g., OBD-II port, USB port, and others) to gain direct access to the internal network. The owner or the driver has the advantage of log inside the vehicle for an unlimited time.
- Remote access: Other attacks do not require any direct access to the target vehicle. Attackers could target the vehicle remotely. Such attacks take advantage of the integrated wireless features of modern cars. These features include Bluetooth, a cellular connection, wireless tire pressure monitoring, etc. Attackers need to be within a particular distance of the targeted vehicle; this distance is based on the technology used to attack the vehicle, e.g., 40 meters for wireless communication. Long-range wireless technologies give the attacker the ability to target the system from very far away, as in the case of using the entertainment system to play a song laced with malware which is able to emit malicious messages to the CAN bus [26].
- Mixed access: Direct access to the vehicle could be a means to introduce remote attacks. Indeed, some attackers with rapid direct access to the vehicle may install devices inside the vehicle (such as a USB cover, malicious DVD, malicious component connected via OBDII port, etc.) or outside it (communication sniffing devices). Later, they can employ those parasitic devices to target the vehicle remotely. Attackers may use other people to install these devices, such as a valet who parks the victim's car, a mechanic at a service station [26], and so on.

3.6. Abstract Model and General Attack Trees

In this subsection, we demonstrate how to apply the last step of adopting the SAVTA threat model. The outcome of this step is classification of as many known threats against vehicular systems as possible. Such a classification could reduce redundancy and inconsistency when applying defense techniques against homogeneous threats. In addition, it provides the basis for defining general attack trees. Three layers were used to identify and classify threats and create the general attack trees (see Figure 4):

- Targeted assets: the first layer of the model contains all the (sub)system assets (e.g., hardware, software and firmware, data, or surrounding infrastructure). Within each of these assets, there are different vulnerabilities which could be targeted by motivated attackers.
- Requirements violation: the exploitation of an existing vulnerability in any asset will lead to a violation of one or more of the security requirements (i.e., confidentiality, integrity, availability, authorization, and authentication). We can further identify and classify potential threats based on the violated requirement(s).
- Accessibility: eventually, the way of accessing the assets (i.e., remote, direct, or mixed access) in order to exploit a specific vulnerability is used as the last level for compartmentalization.

Applying this model to the entire vehicle system will identify all the known threats. For each asset a minimum list of 15 general threats needs to be checked. Each of these threats is used to create the root of a general attack tree which explains how an attacker could cause that threat. For example, disabling one of the vehicle sensors is the root of one general attack tree. Disabling such a sensor requires the existence of a specific vulnerability in that sensor which could prevent it from functioning if it is exploited.

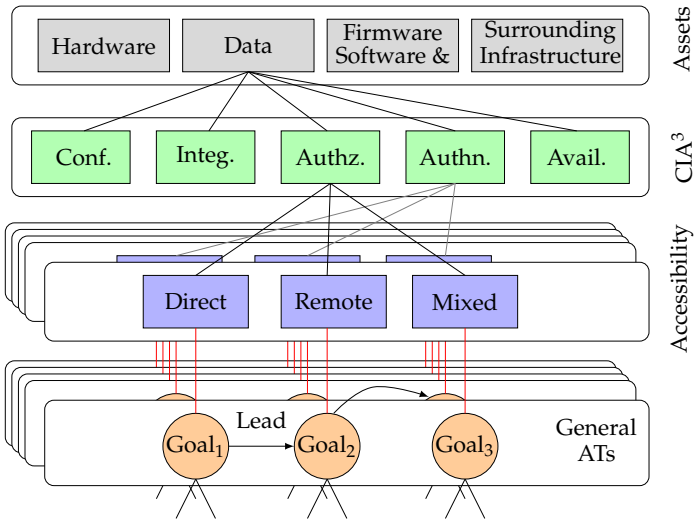

Figure 4. SAVTA method procedure to identify threats and link them to general attack trees.

Building the attack tree will help to illustrate attack vectors which threaten particular vehicle (sub)systems. These trees will turn into distinct ones, gradually reflecting the various studied subsystems. The accomplishment of one tree could open the door to fulfillment of other trees, as we explained within the stepping-stone attack. However, general attack trees seem to be indispensable for avoiding redundancy and interference between the high number of integrated sub-systems within the vehicle. The general trees will be derived from threats which were identified by our proposed threat model.

4. Attack Tree and Risk Analysis

Attack trees have been used to evaluate the security risk to the system and calculate the difficulty and the probability of a successful attack. Determining the attack probability (i.e., likelihood) is related to the difficulty of identifying and exploiting attack scenarios. The great difficulty in performing an attack leads to a low probability of this attack occurring. This difficultly is dependent on many aspects (see Table 2) such as the time required for an attack, the desired attack tools, knowledge of system, and so forth [42,69,70]. However, regarding the risk analysis within the vehicular domain, calculation of the probability of potential attacks based on associating numeric values with each level of these factors, may need to be reconsidered.

Table 2. Factors for calculating the attack difficulty based on [69,70].

Factor	Description
Elapsed Time	The required time to identify a vulnerability in one asset and exploit it
Expertise	The level of attacker expertise (i.e., layman, proficient, expert, or collaboration of many experts)
Opportunity	the window of opportunity to perform the attack
Equipment and tools	The type of equipment and tools which is required to identify and exploit a vulnerability

Elapsed time, for example, has a different effect in terms of the method of carrying out the attack (whether it is a remote attack or one involving direct access to the vehicle). Moreover, the overlap between expertise and tools employed also has different effects. Even inexpert attackers can launch an attack using sophisticated tools. Eventually, stepping-stone attacks should be considered during calculation of the probability of an attack. An attack might be unlikely, but achieving one attack goal in a different subsystem could increase its probability. For example, compromising the internal network of the vehicle could have a high difficulty level. However, the same attack become easier if we consider malicious devices attached by a valet while cleaning the car.

Furthermore, the motivation behind the attack should be considered when calculating the attack severity. The same attack could have varying results. Consider the following three attacks: (1) the driver re-flashes the ECU firmware (or replaces it with another type) to give the vehicle a more powerful performance. (2) A company which uses malicious firmware in one ECU to degrade the performance of a vehicle component or even lead it to produce misleading results intentionally (e.g., Volkswagen's emissions scandal [71]). (3) A hacker or terrorist who manages to compromise the firmware of an ECU to steal the driver's information or to cause an accident. In all three cases, the attack is the same (i.e., compromising the ECU firmware) while the motivation and severity are different.

Finally, using risk assessment to identify the riskiest (sub)system and apply mitigation for that specific (sub)system is not enough to secure the vehicle. Many authors have argued that mitigating all the system vulnerabilities may involve significant costs, and therefore, particular vulnerabilities which have a low risk should be ignored, with the system accepting that risk. In any case, an attacker needs only one security vulnerability to break the entire system. Therefore, attackers will ignore highly protected (sub)systems and move to others which are not protected. Thus, the evaluation of the attack should play a role in determining the reaction to it but not whether or not we should mitigate it.

5. Use Case: Autonomous Vehicle Driving

Figure 5 depicts the simplified hardware architecture of the autonomous vehicle driving system. Three types of hardware components are used within this use-case: a) smart sensors (e.g., GPS, Lidar, etc.) which are used to gather data about the environment and deliver it to b) many ECUs running the different autonomous vehicle software systems which emit control commands to the c) actuators (e.g., steering, throttle, brakes, etc.) which apply those commands. The autonomous vehicle software components can be categorized into four main subsystems [72]: (1) Localization: refers to the ability of the vehicle to determine its position concerning the surrounding environment as well as to get a good estimation of road traveled. These position information are fed by GPS and inertial data via CAN bus to the ECU which is responsible for vehicle localization and motion estimation. (2) Perception: translates the received raw sensor data about the surrounding environment into useful information used to obtain a safe trajectory. Lidar sensors and a camera are streamed to the ECU which is responsible for environment perception (sensor data processing, data fusion, environment modeling). Data from a radar sensor is acquired via a CAN bus connection. (3) Planning: the main aim of this subsystem is utilizing aggregated data, which is provided by the perception subsystems to plan actuation of the vehicle. This includes the optimal trajectory planning, behavioral planning, motion planning, etc. The planner unit passes the ultimate information/commands to the control unit. (4) Controlling: this subsystem receives the ultimate information/commands of the planner unit and passes them to the actuators which generate the desired actions such as increasing the speed or moving the steering and so forth. These components collaborate to support different safety and non-safety systems such as Adaptive Cruise Control and Automated Obstacle Avoidance systems.

We used our abstract model to identify the potential threats within the automated obstacle avoidance use case, and to show how vulnerability or attack on surrounding environment component may lead to wrong planning of the autonomous vehicle. Firstly, we need to define all components which could include vulnerabilities. We concentrate on the hardware components which are used within our use case. LiDAR, Camera, Radar, and GPS are possible (hardware) attack surfaces which can be used to target the vehicle (see Figure 5). To keep the analysis simple, we decide to focus on one of these hardware components, which is the camera in this case. We also consider some of the surrounding infrastructures which is captured by the camera. In the same time, we focused on the data after it was delivered to the ECU which host the environment perception component. Note that we did not consider the software component which processes the captured image, the ECU where this component is mapped , or the transferred data between ECU_1 and ECU_2.

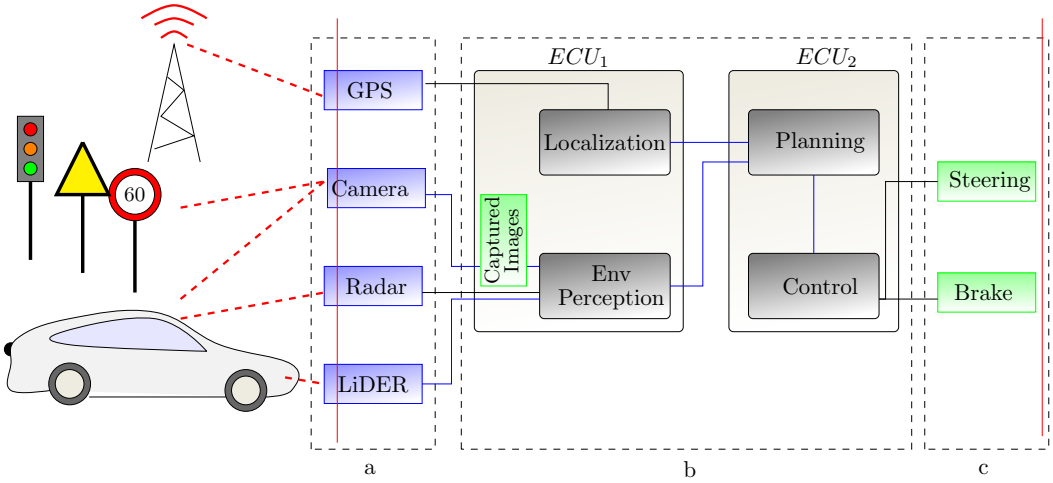

Figure 5. Simplified architecture of Automated Obstacle Avoidance use case which contains (**a**) smart sensors, (**b**) Electronic Control Units (ECUs), (**c**) actuators, and the surrounding infrastructure.

Figure 6 shows the general attack trees for the camera, the road sign, and the captured images. Also, the figure presents the threat agents who may target the system based on the abstract model and the analysis that we performed. Table 3 contains the meaning for each node of our attack tree.

Table 3. Notations and meanings of nodes.

Notion	Meaning	Notion	Meaning
G_1	Disabling the camera	$G_{3.2}$	Distorting the road sign [66]
$G_{1.1}$	Blinding the camera [57]	G_4	Manipulating the road signs
$G_{1.1.1}$	Placing a LED device	$G_{4.1}$	Changing the data on the road sign
$G_{1.1.2}$	Emitting a strong light to the camera	$G_{4.2}$	Installing a fake sign
$G_{1.2}$	Jamming Attack [57]	$G_{4.3}$	Projecting images of fake road signs [73]
$G_{1.3}$	Covering the camera with tape	G_5	Manipulating the stored images
$G_{1.4}$	Breaking the camera	$G_{5.1}$	Manipulating the ECU_1
G_2	Confusing proper function of the camera	$G_{5.2}$	Changing the stored image or writing fake bits
$G_{2.1}$	Confusing the auto controls of the camera	G_6	Disclosing the captured images
G_3	Disabling road signs	$G_{6.1}$	Extracting the images and transmitting them to remote location
$G_{3.1}$	Removing the road sign totally		

As we have described in Section 3.6, after choosing an asset, we need to check the security requirements of the CIA^3 that could be violated by attackers directly or remotely when they target the selected asset. For the camera, we found that Integrity and Availability are the most targeted requirements. Attackers can disable the camera remotely (G_1) using different kinds of attacks such as a blinding attack ($G_{1.1}$) or jamming attack ($G_{1.2}$) as illustrated in [57]. The blinding attack can be performed by placing an LED device in a suitable place on the road ($G_{1.1.1}$) and using this device to emit intense light into the camera ($G_{1.1.2}$) (PAND gate was used to link these tow nodes. See Figure 6). This overexposed light prevents the camera from detecting the objects on the road. Performing such attacks requires an adequate level of skills and resources. The available evidence showed that security experts had performed such attack for research purposes. Also, disabling the camera can occur directly by covering the camera with dark tape ($G_{1.3}$) or even wrecking it ($G_{1.4}$). Confusing the functionality of the camera (G_2) could be another goal of the attacker, which could happen by damaging its auto control system ($G_{2.1}$). Such an attack does not require any sophisticated tools or skills. At the same time, it is easy to be detected.

The attackers could target the recorded data by the camera. This data need to be protected since it may include sensitive information about the other vehicles or the pedestrians around the car [74]. An attacker may try to extract this data and use it in a way that violates the privacy of the pedestrians or the other users of the street ($G_{6.1}$). Attackers also may try to manipulate the captured data ($G_{5.1}$) to cause a wrong environment perception. Both attacks require controlling of the ECU where the captured imaged is stored and processed ($G_{5.2}$).

Another case that could affect the integrity of the camera data is when the captured objects, such as road signs, traffic lights, and so forth, are mangled or removed. These components belong to the surrounding environment. We concentrate on the road signs component only. We determine the security requirements which could be violated as a result of attacks against these signs. The most common attacks which target road signs are removing ($G_{3.1}$) or distorting them ($G_{3.2}$) as in [66]. Car thieves could perform such attacks. Some attackers could try to change the information on the signs ($G_{4.1}$) or even install new fake signs ($G_{4.2}$). Another exciting attack is the one in which an attacker used a drone equipped with an image projector to project fake road signs ($G_{4.3}$) as explained in [73].

In Figure 6, we used the "lead" arc to show how an attacker is able to gain the goal of an attack tree without the need for using any of its sub-trees. E.g., manipulation of the surrounding infrastructures has a direct effect on the functionality of different components in our use case. As a result of such attacks, the camera will deliver incorrect information to the perception module, that causes improper planning and motions. The wrong planning process may cause the car to drive to unwanted places; for example, an area where the thief is waiting.

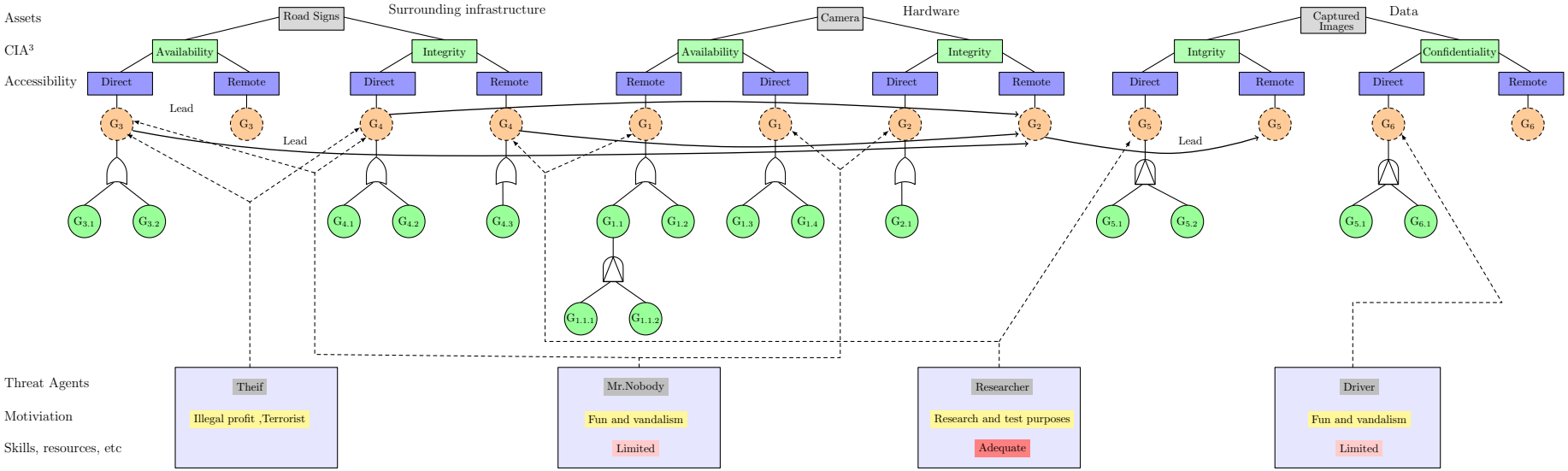

Figure 6. General attack trees for the camera, road sign, and captured images in our use-case.

6. Conclusions

In this paper, we created a comprehensive threat model based on the existing vehicle-related threat modeling efforts. The model tries to answer three main questions: (1) what are we facing? By defining threat agents who may target the vehicle system and their motivation and tools. (2) What do we have? By classifying the different assets (e.g., hardware, software) and their possible threats. (3) What do we need? By defining the security requirements of the defined assets. Our model classifies and identifies the threats based on targeted assets, the violated security requirements, and the accessibility of the threats. General attack trees can be linked to each of the identified threats.

Having a tool-chain that can support the implementation of SAVTA is one important point that we want to consider as future work. We plan to adopt the NCC Group Automotive Threat Modeling template and extend it to include all the missing aspects of the SAVTA model. Another plan is to perform an extensive evaluation with respect to the costs and time necessary to perform the SAVTA model. The quantitative analysis of the attacker tree is not included in our current research. Calculating the probability and importance of general attacks tress [75,76] are examples of what we intend to research as future work. Besides, extending the generated attack trees to create attack–defense trees [77] or attack–countermeasure trees [78] is another plan for future work.

Author Contributions: Conceptualization, M.H.; writing—original draft preparation, M.H.; writing—review and editing, M.H. and V.P.; supervision, V.P.; project administration, V.P.; funding acquisition, V.P. All authors have read and agreed to the published version of the manuscript.

Funding: This work is partially supported by the European Commission through the following H2020 projects: THREAT-ARREST under Grant Agreement No. 786890, I-BiDaaS under Grant Agreement No. 780787, CONCORDIA under Grant Agreement No. 830927, C4IIoT under Grant Agreement No. 833828, and SmartShip under Grant Agreement No. 823916.

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

References

1. Broy, M.; Kruger, I.H.; Pretschner, A.; Salzmann, C. Engineering automotive software. *Proc. IEEE* **2007**, *95*, 356–373. [CrossRef]
2. Charette, R.N. This car runs on code. *IEEE Spectr.* **2009**, *46*, 3.
3. Wolf, M.; Weimerskirch, A.; Paar, C. Secure in-vehicle communication. In *Embedded Security in Cars*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 95–109.
4. Tuohy, S.; Glavin, M.; Hughes, C.; Jones, E.; Trivedi, M.; Kilmartin, L. Intra-vehicle networks: A review. *IEEE Trans. Intell. Transp. Syst.* **2014**, *16*, 534–545. [CrossRef]
5. Miller, C.; Valasek, C. Remote exploitation of an unaltered passenger vehicle. *Black Hat USA* **2015**, *2015*, 91.
6. Link, R. Is Your Car Broadcasting Too Much Information? 2015. Available online: https://blog.trendmicro.com/trendlabs-security-intelligence/is-your-car-broadcasting-too-much-information/ (accessed on 18 May 2020).
7. Fabian A. Scherschel, D.S. Beemer, Open Thyself!—Security vulnerabilities in BMW's ConnectedDrive. 2015. Available online: https://www.heise.de/ct/artikel/Beemer-Open-Thyself-Security-vulnerabilities-in-BMW-s-ConnectedDrive-2540957.html (accessed on 18 May 2020).
8. Lodge, D. Hacking the Mitsubishi Outlander PHEV Hybrid. 2016. Available online: https://www.pentestpartners.com/security-blog/hacking-the-mitsubishi-outlander-phev-hybrid-suv/ (accessed on 18 May 2020).
9. Thompson, C. A Hacker Figured Out a Way to Almost Completely Control GM Cars with OnStar. 2015. Available online: https://www.businessinsider.com/hackers-device-can-take-over-gm-cars-with-onstar-system-2015-7?IR=T (accessed on 18 May 2020).
10. SAE Vehicle Electrical System Security Committee. *Sae j3061-Cybersecurity Guidebook for Cyber-Physical Automotive Systems*; SAE-Society of Automotive Engineers: Warrendale, PA, USA, 2016.
11. Schneier, B. Attack Trees - Modeling security threats. *Dr. Dobb's J.* **1999**, 24, 21–29.
12. Shirey, R.W. *Internet Security Glossary*, version 2; RFC 4949; 2007; Available online: https://www.rfc-editor.org/info/rfc4949 (accessed on 18 May 2020).

13. International Organization for Standardization. *Information Technology—Security Techniques—Information Security Management Systems—Overview and Vocabulary*; Standard, International Standard ISO 27000; International Organization for Standardization: Geneva, Switzerland, 2016.
14. Shostack, A. *Experiences Threat Modeling at Microsoft*; MODSEC@MoDELS. 2008. Available online: https://adam.shostack.org/modsec08/Shostack-ModSec08-Experiences-Threat-Modeling-At-Microsoft.pdf (accessed on 18 May 2020).
15. Shostack, A. *Threat Modeling: Designing for Security*; John Wiley & Sons, Inc.: Indianapolis, IN, USA, 2014.
16. Casey, T. *Threat Agent Library Helps Identify Information Security Risks*; Intel Corporation White Paper; Intel Corporation: Santa Clara, CA, USA, 2007; Volume 2; Available online: https://www.sbs.ox.ac.uk/cybersecurity-capacity/system/files/Intel%20-%20Threat%20Agent%20Library%20Helps%20Identify%20Information%20Security%20Risks.pdf (accessed on 18 May 2020).
17. Rosenquist, M. *Prioritizing Information Security Risks with Threat Agent Risk Assessment*; Intel Corporation White Paper. 2009; Available online: https://media10.connectedsocialmedia.com/intel/10/5725/Intel_IT_Business_Value_Prioritizing_Info_Security_Risks_with_TARA.pdf (accessed on 18 May 2020).
18. Hamad, M.; Nolte, M.; Prevelakis, V. Towards Comprehensive Threat Modeling for Vehicles. In Proceedings of the 1st Workshop on Security and Dependability of Critical Embedded Real-Time Systems, Porto, Portugal, 28 November 2016; p. 31.
19. Camek, A.G.; Buckl, C.; Knoll, A. Future Cars: Necessity for an Adaptive and Distributed Multiple Independent Levels of Security Architecture. In Proceedings of the 2nd ACM International Conference on High Confidence Networked Systems, HiCoNS '13, Philadelphia, PA, USA, 8–13 April 2013; ACM: New York, NY, USA, 2013; pp. 17–24. [CrossRef]
20. Bezemskij, A. Detecting Cyber-Physical Threats Against Autonomous Robotic Systems in Routine Missions. Ph.D. Thesis, University of Greenwich, London, UK, 2017.
21. Karahasanovic, A.; Kleberger, P.; Almgren, M. Adapting Threat Modeling Methods for the Automotive Industry. In Proceedings of the 15th ESCAR Conference, Berlin, Germany, 7–8 November 2017; pp. 1–10.
22. Caralli, R.A.; Stevens, J.F.; Young, L.R.; Wilson, W.R. *Introducing Octave Allegro: Improving the Information Security Risk Assessment Process*; Technical Report; Software Engineering Inst., Carnegie-Mellon Univ.: Pittsburgh, PA, USA, 2007.
23. ERSI. *Intelligent Transport Systems (ITS); Security; Threat, Vulnerability and Risk Analysis (TVRA)*; Technical Report; ETSI: Sophia Antipolis, France, 2010.
24. Skybox™ Security. Threat-Centric Vulnerability Management (TCVM). 2019. Available online: https://www.infosecurityeurope.com/__novadocuments/480016?v=636628566546630000 (accessed on 18 May 2020).
25. 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, 8–12 August 2011.
26. 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.
27. Kohnfelder, L.; Garg, P. The Threat to our Products. 1999. Available online: https://adam.shostack.org/microsoft/The-Threats-To-Our-Products.docx (accessed on 18 May 2020).
28. Winsen, S. Threat Modelling for Future Vehicles: On Identifying and Analysing Threats for Future Autonomous and Connected Vehicles. Master's Thesis, University of Twente, Enschede, The Netherlands, 2017.
29. Macher, G.; Sporer, H.; Berlach, R.; Armengaud, E.; Kreiner, C. SAHARA: A security-aware hazard and risk analysis method. In Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition (DATE), Grenoble, France, 9–13 March 2015; pp. 621–624.
30. Monteuuis, J.P.; Boudguiga, A.; Zhang, J.; Labiod, H.; Servel, A.; Urien, P. Sara: Security automotive risk analysis method. In Proceedings of the 4th ACM Workshop on Cyber-Physical System Security, Incheon, Korea, 4–8 June 2018; pp. 3–14.
31. NCC Group. The Automotive Threat Modeling Template. 2016. Available online: https://www.nccgroup.trust/uk/about-us/newsroom-and-events/blogs/2016/july/the-automotive-threat-modeling-template/ (accessed on 18 May 2020).
32. Microsoft. Microsoft Threat Modeling Tool. Available online: https://www.microsoft.com/en-us/securityengineering/sdl/threatmodeling (accessed on 18 May 2020).

33. Ma, Z.; Schmittner, C. Threat modeling for automotive security analysis. *Adv. Sci. Technol. Lett.* **2016**, *139*, 333–339.
34. Lautenbachl, A.; Islam, M. Security models. Deliverable D2: HEAVENS. HEAling Vulnerabilities to ENhance Software Security and Safety. 2016. Available online: https://autosec.se/wp-content/uploads/2018/03/HEAVENS_D2_v2.0.pdf (accessed on 18 May 2020).
35. Moore, A.; Ellison, R.; Linger, R. *Attack Modeling for Information Security and Survivability*; Technical Report CMU/SEI-2001-TN-001; Software Engineering Institute, Carnegie Mellon University: Pittsburgh, PA, USA, 2001.
36. Arnold, F.; Guck, D.; Kumar, R.; Stoelinga, M. Sequential and parallel attack tree modelling. In *International Conference on Computer Safety, Reliability, and Security*; Springer: Cham, Switzerland, 2015; pp. 291–299.
37. Vesely, W.E.; Goldberg, F.F.; Roberts, N.H.; Haasl, D.F. *Fault Tree Handbook*; Technical Report; Nuclear Regulatory Commission: Washington, DC, USA, 1981.
38. Izosimov, V.; Asvestopoulos, A.; Blomkvist, O.; Törngren, M. Security-aware development of cyber-physical systems illustrated with automotive case study. In Proceedings of the 2016 Design, Automation & Test in Europe Conference & Exhibition, DATE 2016, Dresden, Germany, 14–18 March 2016.
39. Nigam, V.; Pretschner, A.; Ruess, H. Model-Based Safety and Security Engineering. *arXiv* **2018**, arXiv:1810.04866.
40. Kong, H.K.; Hong, M.K.; Kim, T.S. Security risk assessment framework for smart car using the attack tree analysis. *J. Ambient Intell. Humaniz. Comput.* **2018**, *9*, 531–551. [CrossRef]
41. Hamad, M.; Tsantekidis, M.; Prevelakis, V. Red-Zone: Towards an Intrusion Response Framework for Intra-Vehicle System. In Proceedings of the 5th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS), Crete, Greece, 3–5 May 2019.
42. Henniger, O.; Apvrille, L.; Fuchs, A.; Roudier, Y.; Ruddle, A.; Weyl, B. Security requirements for automotive on-board networks. In Proceedings of the 2009 9th International Conference on Intelligent Transport Systems Telecommunications (ITST), Lille, France, 20–22 October 2009.
43. Ruddle, A.; Weyl, B.; Idrees, S.; Roudier, Y.; Friedewald, M.; Leimbach, T.; Fuchs, A.; Gürgens, S.; Henninger, O.; Rieke, R.; et al. Security Requirements for Automotive On-Board Networks Based on Dark-Side Scenarios. Deliverable D2.3: EVITA. E-Safety Vehicle Intrusion Protected Applications. 2009. Available online: https://www.researchgate.net/publication/46307752_Security_requirements_for_automotive_on-board_networks_based_on_dark-side_scenarios_Deliverable_D23_EVITA_E-safety_vehicle_intrusion_protected_applications (accessed on 18 May 2020).
44. Aijaz, A.; Bochow, B.; Dötzer, F.; Festag, A.; Gerlach, M.; Kroh, R.; Leinmüller, T. Attacks on inter vehicle communication systems-an analysis. In Proceedings of the 3rd International Workshop on Intelligent Transportation (WIT 2006), Hamburg, Germany, 14–15 March 2006.
45. McCarthy, C.; Harnett, K.; Carter, A. *Characterization of Potential Security Threats in Modern Automobiles: A Composite Modeling Approach*; Technical Report; National Highway Traffic Safety Administration: Washington, DC, USA, 2014.
46. Mead, N.R.; Shull, F.; Vemuru, K.; Villadsen, O. *A Hybrid Threat Modeling Method*; Technical Report-CMU/SEI-2018-TN-002; Carnegie Mellon University-Software Engineering Institute: Pittsburgh, PA, USA, 2018.
47. Von Clausewitz, C.; Howard, M.E.; Paret, P. *On War*; Princeton University Press:Princeton, NJ, USA, 1984.
48. Stevens, R.; Votipka, D.; Redmiles, E.M.; Ahern, C.; Sweeney, P.; Mazurek, M.L. The Battle for New York: A Case Study of Applied Digital Threat Modeling at the Enterprise Level. In Proceedings of the 27th USENIX Security Symposium (USENIX Security 18), Baltimore, MD, USA, 15–17 August 2018; USENIX Association: Baltimore, MD, USA, 2018; pp. 621–637.
49. Anderson, R. On the security of digital tachographs. In *European Symposium on Research in Computer Security*; Springer: Berlin/ Heidelberg, Germany, 1998; pp. 111–125.
50. Meredith, R. VW agrees to pay G.M. $100 million in Espionage Suit. 1997. Available online: https://www.nytimes.com/1997/01/10/business/vw-agrees-to-pay-gm-100-million-in-espionage-suit.html (accessed on 18 May 2020).
51. Poulsen, K. Hacker Disables More Than 100 Cars Remotely. 2010. Available online: https://www.wired.com/2010/03/hacker-bricks-cars/ (accessed on 18 May 2020).
52. Nimmo, K. Richard Clarke: Hastings Accident "Consistent with a Car Cyber Attack". 2013. Available online: http://www.informationliberation.com/?id=44269 (accessed on 18 May 2020).

53. Kocher, P.C. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In *Annual International Cryptology Conference*; Springer: Berlin/Heidelberg, Germany, 1996; pp. 104–113.
54. Saeedi, E.; Kong, Y. Side-channel vulnerabilities of automobiles. *Trans. IoT Cloud Comput.* **2014**, *2*, 1–8.
55. Eisenbarth, T.; Kasper, T.; Moradi, A.; Paar, C.; Salmasizadeh, M.; Shalmani, M.T.M. On the power of power analysis in the real world: A complete break of the KeeLoq code hopping scheme. In *Annual International Cryptology Conference*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 203–220.
56. 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**, *16*, 993–1006. [CrossRef]
57. 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.
58. Shin, H.; Kim, D.; Kwon, Y.; Kim, Y. Illusion and dazzle: Adversarial optical channel exploits against lidars for automotive applications. In *International Conference on Cryptographic Hardware and Embedded Systems*; Springer: Cham, Switzerland, 2017, pp. 445–467.
59. Wasicek, A.; Andre, W. Recognizing Manipulated Electronic Control Units. In Proceedings of the SAE 2015 World Congress & Exhibition, Detroit, MI, USA, 21–23 April 2015.
60. Yoney, D. Tesla Model S Owners Hack Their Cars, Find Ubuntu.2014. Available online: https://www.autoblog.com/2014/04/12/tesla-model-s-owners-hack-their-cars-find-ubuntu/ (accessed on 18 May 2020).
61. Dunn, M. Toyota's killer firmware: Bad design and its consequences. *EDN Network.* **2013**. Available online: http://faculty.cs.tamu.edu/ioerger/ethics/Toyota-s-killer-firmware--Bad-design-and-its-consequences-1.pdf (accessed on 18 May 2020).
62. Bécsi, T.; Aradi, S.; Gáspár, P. Security issues and vulnerabilities in connected car systems. In Proceedings of the 2015 International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS), Budapest, Hungary, 3–5 June 2015, pp. 477–482.
63. Wasicek, A.; Weimerskirch, A. *Recognizing Manipulated Electronic Control Units*; SAE Technical Report; SAE: Warrendale, PA, USA, 2015.
64. Bogage, J. Scary Glitch Affects Luxury Cars. 2016. Available online: https://www.bostonglobe.com/lifestyle/2016/06/09/scary-glitch-affects-luxury-cars/kj4wg2lhphlJDC3gATGuPM/story.html (accessed on 18 May 2020).
65. Rouf, I.; Miller, R.; Mustafa, H.; Taylor, T.; Oh, S.; Xu, W.; Gruteser, M.; Trappe, W.; Seskar, I. Security and Privacy Vulnerabilities of In-car Wireless Networks: A Tire Pressure Monitoring System Case Study. In Proceedings of the 19th USENIX Conference on Security (USENIX Security'10), Washington, DC, USA, 11–13 August 2010; USENIX Association: Berkeley, CA, USA, 2010.
66. 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 visual classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 1625–1634.
67. Olofsson, J. 'Zombies ahead!'A study of how hacked digital road signs destabilize the physical space of roadways. *Vis. Commun.* **2014**, *13*, 75–93. [CrossRef]
68. Verdult, R.; Garcia, F.D.; Ege, B. Dismantling megamos crypto: Wirelessly lockpicking a vehicle Immobilizer. In Proceedings of the Supplement to the 22nd USENIX Security Symposium (USENIX Security 15), Washington, DC, USA, 12–14 August 2015.
69. International Organization of Standardization. *Information Technology–Security Techniques–Methodology for IT Security Evaluation*; Standard, International Standard ISO/IEC 18045; ISO: Geneva, Switzerland, 2008.
70. International Organization for Standardization. *Information Technology – Security Techniques–Evaluation Criteria for IT Security*; Technical Report; International Organization for Standardization: Geneva, Switzerland, 2009.
71. Guilbert, G.; Jack, E.; Karl, R.; Deerek, W. Explaining Volkswagen's Emissions Scandal. *New York Times*, 2016. Available online: https://sit.instructure.com/courses/17250/files/2569242/download?download_frd=1 (accessed on 18 May 2020).
72. Pendleton, S.; Andersen, H.; Du, X.; Shen, X.; Meghjani, M.; Eng, Y.; Rus, D.; Ang, M. Perception, planning, control, and coordination for autonomous vehicles. *Machines* **2017**, *5*, 6. [CrossRef]
73. Nassi, D.; Ben-Netanel, R.; Elovici, Y.; Nassi, B. MobilBye: Attacking ADAS with Camera Spoofing. *arXiv* **2019**, arXiv:1906.09765.
74. Strachan, L.A. Re-mapping privacy law: How the google maps scandal requires tort law reform. *Rich. JL Tech.* **2010**, *17*, 1.

75. Shafiee, M.; Enjema, E.; Kolios, A. An integrated FTA-FMEA model for risk analysis of engineering systems: A case study of subsea blowout preventers. *Appl. Sci.* **2019**, *9*, 1192. [CrossRef]
76. Chybowski, L. Importance Analysis of Components of a Multi-Operational-State Power System Using Fault Tree Models. *Information* **2020**, *11*, 29. [CrossRef]
77. Kordy, B.; Mauw, S.; Radomirović, S.; Schweitzer, P. Foundations of attack–defense trees. In *International Workshop on Formal Aspects in Security and Trust*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 80–95.
78. Roy, A.; Kim, D.S.; Trivedi, K.S. Attack countermeasure trees (ACT): Towards unifying the constructs of attack and defense trees. *Secur. Commun. Netw.* **2012**, *5*, 929–943. [CrossRef]