Resilient Shield: Reinforcing the Resilience of Vehicles Against Security Threats

Resilient Shield: Reinforcing the Resilience of Vehicles Against Security Threats

Kim Strandberg*†, Thomas Rosenstatter*, Rodi Jolak‡, Nasser Nowdehi†, Tomas Olovsson*
*Chalmers University of Technology, Sweden, {firstname.lastname}@chalmers.se
†Volvo Car Corporation, Sweden, {firstname.lastname}@volvocars.com
‡Chalmers | Gothenburg University, Sweden, {firstname.lastname}@cse.gu.se

***Abstract*—Vehicles have become complex computer systems with multiple communication interfaces. In the future, vehicles will have even more connections to e.g., infrastructure, pedestrian smartphones, cloud, road-side-units and the Internet. External and physical interfaces, as well as internal communication buses have shown to have potential to be exploited for attack purposes. As a consequence, there is an increase in regulations which demand compliance with vehicle cyber resilience requirements. However, there is currently no clear guidance on how to comply with these regulations from a technical perspective.**

To address this issue, we have performed a comprehensive threat and risk analysis based on published attacks against vehicles from the past 10 years, from which we further derive necessary security and resilience techniques. The work is done using the *SPMT* methodology where we identify vital vehicle assets, threat actors, their motivations and objectives, and develop a comprehensive *threat model*. Moreover, we develop a comprehensive *attack model* by analyzing the identified threats and attacks. These attacks are filtered and categorized based on attack type, probability, and consequence criteria. Additionally, we perform an exhaustive mapping between asset, attack, threat actor, threat category, and required mitigation mechanism for each attack, resulting in a presentation of a secure and resilient vehicle design. Ultimately, we present the *Resilient Shield* a novel and imperative framework to justify and ensure security and resilience within the automotive domain.

***Index Terms*—cyber resilience, security, vehicular systems, automotive systems**

I. Introduction

The complexity of vehicles is increasing. Consequently, vulnerabilities which might be exploited increase as well. Attacks to vehicular systems can be realized: (i) *indirectly* via compromised devices e.g., phones, dongles, or workshop computers connected to vehicle interfaces; (ii) *directly* via physical interfaces e.g., debug ports and the OBD-II connector; and (iii) *remotely* via various malicious sources, such as rogue access points and compromised servers. It has been demonstrated that vehicle cyber attacks e.g., physical attacks [1] and remote attacks [2] are potential threats that have to be taken seriously. As a case in point, Miller and Valasek [3] performed a successful remote attack on a Jeep Cherokee via the Internet taking control of its primary functions by exploiting an open port via a cellular channel, an attack that led to a recall of 1.4 million vehicles. In [4], researchers managed to get remote access to the CAN bus of a BMW by compromising its infotainment system, allowing them to execute arbitrary diagnostic requests. Vulnerabilities in phone applications paired to vehicles have been exploited by adversaries to track vehicles, unlock the doors and to start their ignitions [5]–[7].

Motivation. Securing a vehicle as an afterthought is cumbersome, considering both the complexity which constantly increases and the existing dependencies on current architectural design. Hence, it is imperative to consider security during the vehicle's complete life cycle from idea to cessation.

There are increased requirements towards ensuring a resilient vehicle design, in a way that a vehicle should be able to withstand various types of cyber attacks, malfunctioning units, and other external disturbances. Consequently, the resilient design should be able to *prevent*, *detect*, and *respond* to cyber attacks, something which is also in line with the UNECE regulation [8] and the upcoming standard for automotive cyber security ISO 21434 [9]. In short, *prevention* is accomplished with security controls, *detection* by identifying faults and attacks, and *response* are mechanisms related to handling the detected anomalies with the ability to restore and maintain operation. However, there is currently no clear guidance how to comply with the aforementioned regulations and standards from a technical perspective. The *start*, *predict*, *mitigate*, and *test* (*SPMT*) is a systematic approach for identification and mitigation of vulnerabilities in vehicles [10]. The aim of *SPMT* is to ultimately enhance the security of vehicles through their entire life cycle. In this paper, we use and extend the *SPMT* methodology to establish an in-depth resilient design model with imperative mitigation mechanisms.

Contributions. By applying the *SPMT* methodology, we performed a comprehensive threat and risk analysis of 52 published attacks against vehicles from the past 10 years. 37 of these attacks were considered significant due to their high risk and were thus further mitigated with imperative security and resilience techniques. In this process, we have developed a *threat model* for securing vehicles by identifying vital vehicle assets and the related potential threat actors, their motivations and objectives. Moreover, we have developed a comprehensive *attack model* created from the analysis of the identified threats and attacks, further filtered and categorized based on attack type and risk criteria related to the probability and consequences of the attack. We present a comprehensive summery of the result from applying the *SPMT* methodology, an exhaustive mapping between asset, attack, threat actor, threat category and resilience mechanism for each attack. Ultimately, we define necessary security and resilience enhancements for vehicles, the *Resilient Shield*, which also validates the effectiveness of the methodology. To the best of our knowledge, our result is both novel and imperative to justify and ensure security and resilience within the automotive domain.

978-1-7281-8964-2/21/$31.00 ©2021 IEEE

II. Related Work

Good practices for security of smart cars [11], *Cyber security and Resilience of smart cars* [12], and *The Cyber security guidebook for cyber physical vehicle systems, SAE J3061* [13], provide guidelines regarding threat and risk assessment. *EVITA* [14] proposed a method for security, safety, and risk analysis of in-vehicle networks, whereas *HEAVENS* [15] proposed a security model based on security objectives from EVITA and security attributes from Microsoft *STRIDE* [16]. Rosenstatter et al. [17] continue with the result from an analysis such as HEAVENS and map the identified security demands to security mechanisms. However, this mapping focuses only on securing the in-vehicle network.

The *SPMT* methodology builds on existing methods, models and security principles that are applicable to different phases in a vehicle's life cycle. By adapting and incorporating relevant parts suitable for the vehicular domain, a comprehensive security and safety enhancement is achieved. Consequently, the *SPMT* methodology covers the vehicles entire life cycle, something which cannot be achieved with existing methodologies [10]. *SPMT* adopts Microsoft's *STRIDE* categorization [16] which enables a mapping of attacks to a category with associated security attributes. Thus, mitigation mechanisms can be considered for the attribute and consequently mitigate more than one attack. Additionally in *SPMT*, a reduction analysis is performed for critical threats by creating attack trees to connect the vulnerability with the threat, i.e., an attacker wanders from a leaf node (condition) to the root of the tree (attacker objective). Consequently, the closer to the root a countermeasure is placed, the more conditions are mitigated. Moreover, some conditions can be attained by more than one attack, hence a countermeasure can mitigate several attacks.

The REMIND framework [18] for vehicular systems provides a taxonomy for resilience techniques identified from a review of existing work. In this paper we take advantage of previous knowledge and new results by applying the *SPMT* methodology. In the next sections we present the detailed approach followed by the results.

III. Approach

We use the aforementioned *SPMT* model to perform a comprehensive threat modelling and risk assessment of published attacks to further map these threats and attacks to imperative security and resilience mechanisms.

The *SPMT* methodology has 4 phases: *Start*, *Predict*, *Mitigate* and *Test*. In this paper, we perform the first three phases on a Target Of Evaluation (ToE) and analyze security threats and attacks as well as provide mechanisms for the mitigation thereof (see Figure 1).

In the *Start Phase*, we address the following questions. *What are the threats requiring a resilient design? What are the entry points to the vehicle? Who are the actors, their motivators, and their objectives?* The outcome of the *Start Phase* is a threat model and high-level goals for the enforcement of security and safety attributes.

In the *Predict Phase*, we address the following question. *What are the potential attacks?* The outcome of the *Predict Phase* is an *attack model* which contains relevant attacks categorized and filtered according to a stated criteria.

In the *Mitigate Phase*, we address the following question. *What are the needed mechanisms to ensure a resilient design?* The outcome of the *Mitigate Phase* is a resilient design framework i.e., the *Resilient Shield*, which provides mechanisms and goals for detecting, preventing, and responding to security threats and attacks.

The *Test Phase* includes the implementation of the mitigation mechanisms followed by an execution of different security tests, such as fuzz, vulnerability, and penetration testing. In this paper, we do not perform the *Test Phase*; however, we plan to test the identified mitigation mechanisms within an industrial context in the future.

In the following sections, we perform and provide the outcomes of the first three phases of the *SPMT* methodology (see Figure 1) that are used to establish the *Resilient Shield*.

IV. Threat Model

A threat model is created by considering: (i) the target of evaluation (ToE), and (ii) attackers as well as their motivators and objectives. First, our ToE is stated as the complete vehicle provided by the manufacturer, where we propose to include the following assets. As shown in Table I, the relevance of these assets is verified by the mapping to attacks.

Internal and external communication: *Automotive Bus technologies*, e.g., CAN, FlexRay, LIN, MOST and Ethernet. *Connection interfaces*, e.g., OBD-II, USB, debug ports, Wi-Fi and Bluetooth.

Hardware: *ECUs*, e.g., sensor signal processing. *Sensors*, related to speed, position, temperature, airbag and object detection. *Actuators*, translate signals from ECUs into actions, e.g., braking, steering and engine control.

Software in transit, rest or running: *Software update systems*, e.g., over-the-air or workshop updates. *Software installed or running* in ECUs.

Data Storage: *Sensitive data*, e.g., cryptographic keys, forensics logs and reports.

Second, we propose a simplification of threat actors (i.e., attackers) inspired by the work of Karahasanovic et al. [19] in relation to motivators and objectives.

Actors and Motivators. *The Financial Actor* is driven by financial gain in relation to a company (intellectual property), organization or individual. This actor can be the owner who

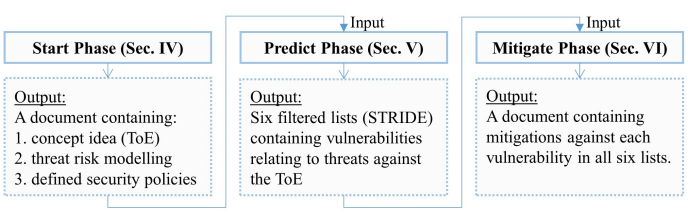

Fig. 1. The first three phases of the SPMT methodology

wants to make unauthorised modifications (e.g., chip tuning) or criminals who install ransomware. *The Foreign Country* is driven by power through cyber warfare, with the intent to disable viable assets within infrastructure (e.g., transportation). *The Cyber Terrorist* is driven by ideological, political or religious objectives. *The Insider* is motivated by retaliation or other personal gains, has knowledge of sensitive information and may plant malicious code into the vehicle. *The Hacktivist* is driven by publicity or adrenaline (i.e., the rush) and can have an agenda for political or social change. *The Script Kiddie* has usually no clear objective, possess limited knowledge and is often using already available tools and scripts. However, the reality is usually a combinations of the mentioned categories and objectives, and actors can be *black hat*, *gray hat*, or *white hat* hackers in relation to society's interpretations of the hackers' intentions. *White hat*, are assumed to be the good guys, *black hats* are the bad guys, and *grey hat* are somewhere in the middle.

Furthermore, in Section VI we adopt the security and safety attributes used in *SPMT*. These attributes are imperative to uphold to ensure a secure and resilient vehicle. On the other hand, the actors are driven by stated *motivators* (e.g., financial, ideological, publicity) with the goal of compromising these attributes. A discussion and a brainstorming about fulfilment of these attributes is part of the *Start Phase*, however we have chosen to include it in Section VI to have all considerations for mitigation in one section. Stated assets and actors are applied to Table I and used in the following section.

V. Attack Model

We perform a qualitative risk assessment of published attacks covered in news media and research publications by estimating (i) the probability and (ii) the consequences of the attacks based on the following criteria. As shown in Table I, the affected assets, the threat actors and the STRIDE categories for each attack are considered during this assessment.

Attack Probability. The first step in this phase is to define attack probability where the three following estimates should be used:

E1: *Where, when, and in what situation can the attack be carried out?*

E2: *What expertise is required of the attacker?*

E3: *How much time does it take to perform the attack?*

The resulting probability is on a scale of 1 to 3, where 3 indicates that an attack is more probable to take place. The highest value in E1-E3 is chosen.

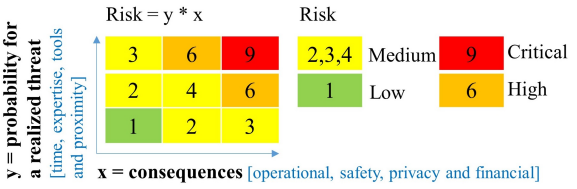

Fig. 2. Adapted table for the risk calculation from the SPMT methodology.

Attack Consequence. In the second step, the consequences are defined by assessing the effect of the attack on the operational, safety, privacy, and financial aspects. The resulting consequence is on a scale from 1 to 3, where 3 indicates that the consequence is more severe. The highest value is chosen.

Risk Assessment. Once we get the estimates of the attack probability and consequences, we estimate the overall risk by calculating the product of the probability and the consequence, which gives a risk value between 1 and 9 (see Figure 2). To achieve a realistic balance between the financial cost for mitigation and its related complexity versus the risk and asset value, we consider only the most significant threats. These threats have a risk value of 6 or 9, which is in line with ISO 26262 and ASIL [20] and corresponds to high and critical risk.

A. Disclosed Attacks

To create the *attack model*, we follow the *SPMT* recommendation for search criteria and query scopus[1] and Google scholar for academic work, and common vulnerability databases (NVD, CVE) with keywords related to vehicle, attack and STRIDE categories (e.g., spoofing) or related terms (e.g., mitm). Moreover, we do query the Google search engine for media reports on attacks. Next, we classify the attacks according to STRIDE categories, followed by some examples. Attacks are considered and analyzed with respect to probability, consequence and risk within their respective category. Out of a total of 52 published attacks, we have identified 37 high and critical risk attacks which are further considered in this work.

1) Spoofing Attacks - Authenticity, Freshness [5], [21]–[38]. The goal of the attacker is to intercept, hijack, manipulate or replay the communication with a potential remote access persistence. *Security flaws in mobile software*, such as demonstrated in the OwnStar attack [5]. OwnStar intercepts communication after the OnStar user opens the application, whereas the OwnStar device gains the user's credentials. Relay attacks, as in compromise of remote keyless entry systems as well as breaking poor authentication mechanisms [21]–[23]. GNSS spoofing considers broadcasting fake signals over authentic in order to to trick a receiver, with the intention to get a vehicle off course [24]. *In-vehicle protocol spoofing*, can affect safety critical actuators, such as brake, steering and engine control. Protocols themselves might lack inherent mechanisms for security which makes active attacks possible such as malicious drop, modify, spoof, flood and replay of messages.

2) Tampering Attacks - Integrity [2], [4], [36], [38]–[41]. Vulnerable USB/OBD-II dongles or compromised in-vehicle devices can potentially enable a hacker to control the communication. Devices can be compromised in various ways e.g., vulnerabilities in proprietary authentication mechanisms can enable the right to run sensitive diagnostics commands. Brute-force attacks can be used to retrieve cryptographic keys, with

[1]https://www.scopus.com/

potential to upload exploits to ECUs. Physical tampering of ECUs or other connected devices. Manipulated firmware in current ECUs, such as malicious code injection via firmware update. Replacement of ECUs or new devices to eavesdrop/inject messages or to manipulate software, modify or compromise vehicle functions. Vulnerable connected devices such as OBD and USB dongles can potentially provide remote access to individual cars and vehicle fleets [40]. Moreover, in [2] firmware was extracted and reverse engineered, manipulated and injected directly into ECU firmware facilitating persistent and bridging capabilities for attacks.

3) Repudiation Attacks - Non-repudiation, Freshness. An attacker manipulates or removes forensic in-vehicle data, such as GPS coordinates, speed, acceleration and brake patterns, with the intention to hide traces of the attack. Despite our best effort, we did not find attacks which can be clearly mapped to this category; however, this type of attacks will likely be more frequent in the future due to both increased number of attacks and digital forensic investigations.

4) Information Disclosure Attacks - Confidentiality, Privacy [7], [38], [39], [42]–[45]. An attacker may be able to exploit cryptographic keys and consequently decrypt sensitive data by e.g., reverse engineering software with hard-coded keys. Bad routines for handling of replaced unit led to leaked sensitive data such as owners home and work address, calendar and call entries and Wi-Fi passwords [42]. A mobile application for vehicle control contained hard-coded credentials, thus an attacker may be able to retrieve sensitive data remotely by recovering the key from the application [7]. A vulnerability in an OBD-II dongle exposed all transferred data to the public [43]. Vulnerabilities in automotive bus technologies make various attacks possible, such as sniffing of CAN traffic due to its broadcast transmission and lack of encryption [44].

5) Denial of Service (DoS) Attacks - Availability [34]–[37], [46]–[49]. Many attacks focus on the in-vehicle network that uses CAN as this technology suffers from fundamental vulnerabilities with respect to security (e.g., broadcast communication, lack of encryption/authentication). Other attacks range from sending an indefinite amount of data to ECUs to make them unresponsive or crash, exploiting error handling mechanisms, or flooding the network with high priority messages in order to block lower priority messages. A vulnerability in the Bluetooth functionality supported unrestricted pairing without a PIN, thus enabled the potential for sending remote CAN commands affecting safety critical assets [48]. The Bus-off attack made ECUs unresponsive or crash [49]. Murvay et al. [47] managed to disable FlexRay nodes by exploitation of the bus guardian, power saving functionality and by causing loss of synchronization.

6) Elevation of Privilege Attacks - Authorization [3], [7], [36], [38], [39], [41], [50]–[52]. In [36] two Bluetooth vulnerabilities allowed remote code execution with root privileges. Moreover, manipulation of the firmware of the infotainment unit enabled injection of arbitrary CAN messages. In [50], they were able to release the airbag by message injection due to a vulnerable authentication mechanism. Lack of authentication in the NissanConnect app allowed to retrieve personal data by entering an URL with the vehicle identification number [52]. The outcome of this phase is applied to Table I and used in the next phase in the following section.

VI. Resilient Shield

In this section we present the *Resilient Shield* which consists of high-level security goals emphasizing the overall design requirements resulting from an analysis of the threat model (Section IV). We further provide in Section VI-B detailed directives for fulfilling the high-level security goals for resilient vehicles which are based on these goals and the *attack model* (Section V). Table I summarizes the *Resilient Shield*. We list automotive assets, associate them with high risk attacks, potential threat actors and STRIDE threat categories, and link these to suitable security and resilience techniques to show how *Resilient Shield* can be used to mitigate these attacks.

A. High-level Security Goals (SGs)

The following high-level goals are the result of an analysis of the *threat model* detailed in Section IV. Each SG is associated with the relevant safety and security attributes they enforce.

SG.1 Secure Communication. *Integrity*, *authenticity* and, in specific cases, *confidentiality* need to be ensured for communication. *Integrity* and *authenticity* allow to verify the origin of the message and protect the message from being altered during transmission. *Confidentiality* can be achieved through encryption of the message to prevent unauthorized read access. *Freshness*, e.g., via counters or timestamps, can be used to mitigate replay attacks.

SG.2 Readiness. *Availability* to authorized entities under normal circumstances as well as disturbances. Even if an adversary tries to disrupt the information flow, the *integrity* and *availability* of correct information needs to be guaranteed.

SG.3 Separation of Duties is needed to limit access to resources for *authorized* entities only. *Authorization* should be combined with the principle of *least privilege* to limit the number of entities having access to a resource to the minimum.

SG.4 Secure Software Techniques need to provide security features to ensure that the executed software has not been modified by an unauthorized entity (*authenticity*) and that the software does not contain disclosed vulnerabilities.

SG.5 Separation/Segmentation on an architectural or process level is necessary in order to limit access and reduce the severity in case of an intrusion (*availability*). *Isolation* techniques, e.g., process isolation, should be considered where possible.

SG.6 Attack Detection and Mitigation is of utmost importance to enable the system to react and ideally prevent further damage to the system.

SG.7 State Awareness should be ensured with the ability to switch between various operational states, thus providing *reliability* and *maintainability*.

SG.8 Forensics is necessary for post analysis of detected malicious events and accordingly updating access control policies and other preventive measures.

Physical security, such as vehicle locks, alarm system, and protecting infrastructure server rooms should be considered. Components must be extensively tested against requirements separately and when integrated into the vehicle, such as stated in the *SPMT Test Phase*. *SPMT* suggests to use both a qualitative and quantitative assessment; however, we focus on the qualitative assessment as the aim of *Resilient Shield* is to guide the resilient design of automotive systems. Moreover, a reduction analysis of attack trees is suggested to find commonalities in countermeasures; however this is not considered and is thus left as future work.

B. Detailed Directives

In this section, we list detailed techniques and patterns that contribute to the security and resilience of automotive systems based on the identified security goals, *threat* and *attack model* presented in this paper. First, we incorporate the identified patterns from the REMIND framework [18] in *Resilient Shield* and further extend them with security techniques to provide a comprehensive collection of both, security and resilience techniques for automotive systems. Second, we further discuss the security aspects of the identified resilience techniques. Next, we detail these techniques.

Authentication: Message authentication can be achieved through Message Authentication Codes (MACs) or signatures which ensure that the message: (i) is created by the claimed source and (ii) has not been altered during transmission. The authentication of devices can verify that the hardware, e.g., the head unit or a diagnostic device, is legit.

Encryption: Encryption of data ensures the protection of intellectual property, makes it more difficult to reverse engineer software, protects cryptographic material and the privacy of users and forensics data.

Redundancy/Diversity: A voting mechanism is used when comparing the output of two or more redundant systems or software functions. Redundancy increases the resilience against anomalies; however, from a security perspective it must be ensured that the voting process cannot be exploited by an attacker to perform DoS or spoofing attacks.

Access Control: Gateways with firewall capabilities allow filtering of messages between different networks in the vehicle. In addition, host-based firewalls on the ECUs can limit the exposure of open communication ports. Securing physical debug ports is vital to protect against unauthorized exploitation. Access control to resources such as files, computation, and diagnostic commands can be provided by the operating system or by e.g., challenge-response authentication.

Runtime Enforcement: Runtime verification is combined with reactive measures when safety properties are violated [18], [53].

Secure Storage: Cryptographic material needs to be protected against unauthorized modifications and read access. Data can be either stored encrypted in the regular file system or in a protected memory partition.

Secure Boot: A validation of the authenticity and integrity of the firmware to be loaded during the boot process [54].

Secure Programming: Secure programming guidelines such as MISRA C [55] are important to avoid common programming errors. Additionally, trusted execution environments may be necessary for isolating and securing applications.

Secure Software Update: The ability to update software is not only a necessity to improve and extend functionality, it is also essential for security, e.g., to mitigate vulnerabilities. In addition, the update process itself needs to be secure [56], during the distribution and installation process.

Verification & Validation: The *Test Phase* in *SPMT* focuses on the need for security testing and verification of each asset by doing fuzz, vulnerability and penetration testing. In addition to security testing, the verification and validation of functionality and safety is required [10], [18].

Separation: Architectural separation can be achieved through physical separation into smaller networks or through virtualization techniques allowing to allocate resources to specific functions or systems.

Specification-based Detection: Knowledge about abnormal behavior is used to detect anomalies and attempts to exploit known vulnerabilities. It also requires domain knowledge and needs to be updated regularly [18], [57].

Anomaly-based Detection: Is based on defining normal behavior and deviations trigger alerts and has the potential to detect unknown attacks. Anomaly-based detection can be categorized in statistical, information-theoretic, machine learning and localization techniques [18], [57].

Prediction of Faults/Attacks: Predicting the next step or the ultimate goal of an ongoing attack.

Adaptive Response: The function response may be temporarily adapted, e.g., through a model, while under attack [18].

Reconfiguration: Graceful degradation can be used to limit the impact of an attack when preventive measures failed.

Migration: The ability to migrate services to other nodes in order to maintain system functions when under attack [18].

Checkpoint & Rollback: Used to recover the system to a desired state. The state needs to be secured, e.g., through secure logging, to defend against possible attacks that aim at modifying a saved system state [18].

Rollforward Actions: Upon detecting an anomaly or error the system transitions back to the state immediately before the event happened. Similarly to rollback it needs to be ensured that this mechanism cannot be exploited [18].

Self-X: The system needs to be aware of its state and able to switch to other states when anomalies occur [18], [58].

Robustness: Employed mechanisms and functions need to be robust against anomalies [18].

Forensics: Secure logging is used to record events, e.g., detection of an ongoing attack, use of specific services or diagnostics. In addition, events with non-repudiation claims can be used as evidence of a crime.

Table I presents the *Resilient Shield*. Assets with high or critical risk threats are associated with appropriate security and resilience techniques demonstrating the ability of *Resilient Shield* to defend against these threats. For example,

TABLE I

RESILIENT SHIELD. A MAPPING FROM AUTOMOTIVE ASSETS TO IDENTIFIED ATTACKS, POTENTIAL THREAT ACTORS, STRIDE THREAT CATEGORIES AND ULTIMATELY TO APPROPRIATE SECURITY AND RESILIENCE TECHNIQUES, AND SECURITY GOALS (SGS).

Assets targeted by attacks with high or critical risk.
ToE category:subcategory reference

■ Resilience patterns identified in REMIND [18]

Potential Threat Actors
Financial Actor (FA)
Foreign Country (FC)
Cyber Terrorist (CT)
Insider (IN)
Hacktivist (HA)
Script Kiddie (SK)

STRIDE categories
(S)poofing
(T)ampering
(R)epudiation
(I)nformation Disclosure
(D)enial of service
(E)levation of privilege

			(SG.1,8) Authentication	(SG.1) Encryption	(SG.2) Redundancy/Diversity ■	(SG.3) Access Control	(SG.3) Runtime Enforcement ■	(SG.4,8) Secure Storage	(SG.4) Secure Boot	(SG.4) Secure Programming	(SG.4) Secure Software Update	(SG.4) Verification & Validation ■	(SG.5) Separation ■	(SG.6) Specification-based Detection ■	(SG.6) Anomaly-based Detection ■	(SG.6) Prediction of Faults/Attacks ■	(SG.6) Adaptive Response ■	(SG.6) Reconfiguration ■	(SG.6) Migration ■	(SG.6) Checkpoint & Rollback ■	(SG.6) Rollforward actions ■	(SG.7) Self-X ■	(SG.7) Robustness ■	(SG.8) Forensics ■
Hardware																								
sensor:camera [34], [35]	FC, CT, HA	S, D			•												•	•				•	•	
sensor:GNSS [24], [26], [29], [30], [32]	FC, CT, HA	S	•		•										•		•	•				•	•	
sensor:lidar [28], [34]	FC, CT, HA	S, D			•												•	•				•	•	
sensor:ultrasonic [35]	FC, CT, HA	S, D			•												•	•				•	•	
Communication																								
internal:can [40], [44], [46], [47], [49]	FA, FC, CT, IN, HA	S, T, I, D	•	•	•	•	•						•	•	•	•		•	•			•		•
internal:flexray [37]	FA, FC, CT, HA	S, D	•			•	•						•	•	•			•				•		•
external:bluetooth [4], [36]	FC, CT, HA	S, T, D, E	•			•							•											
external:usb [4]	FC, CT, HA	S, T, E	•			•							•											
external:keyfob [22], [23]	HA, SK	S	•			•									•								•	
external:wifi [5], [33]	HA, SK	S, I	•	•		•				•			•											
external:cellular [3], [4], [41], [45], [51], [52]	FC, CT, HA, SK	S, T, I, D, E	•			•							•											
external:obdII [7], [27], [31], [38], [40], [43], [46], [48]	CT, HA	S, T, I, D, E	•			•	•						•	•	•			•		•		•		•
external:debugport [3], [41]	HA, IN	I, E	•			•																		
Software																								
running:state [25]	FC, CT, HA	S, D				•				•				•	•					•	•			•
running:firmware [3]–[5], [33], [36], [39], [41], [45], [51], [52]	FC, CT, HA	S, T, E				•				•	•	•	•	•	•			•				•		•
instorage:update [4], [36], [41]	HA, SK	S, T, E	•	•		•		•	•		•		•	•	•	•	•	•	•					•
instorage:weakcrypto [21], [50], [52]	FC, CT, HA, SK	S, E	•							•													•	
Data Storage																								
crypto:certificates [41]	FC, CT, HA	I		•		•		•	•															
hw:replaced [42]	HA, SK	I	•	•		•																		

hacktivists and insiders are the main threat actors for *communication:external:debugport*, such as JTAG, and needs to be protected with authentication mechanisms combined with access control or, if not possible otherwise, with physical protection (e.g., deactivation).

VII. CONCLUSION

We have performed a comprehensive threat and risk analysis of published attacks against vehicles and derived imperative security and resilience mechanisms by applying the *SPMT* methodology. A *threat model* with vital vehicle assets and related potential threat actors, their motivations and objectives was developed. By an extensive analysis of threats and attacks, further filtered and categorized based on attack type, probability and consequence criteria, an *attack model* was developed based on the remaining high risk attacks. Based on the developed models, a comprehensive mapping between asset, attack, threat actor, threat category, and defense mechanisms was performed for all attacks and is presented in Table I. Table I summarizes the outcomes by applying *SPMT*, i.e. the *Resilient Shield*, a novel framework both justifying and defining imperative security and resilient mechanisms needed in a modern vehicle. Consequently, the *Resilient Shield* can be used as a vital baseline for protection against common security threats and attacks.

We believe our work is imperative for facilitating and guiding the design of resilient automotive systems; however, it still remains to be seen how large the coverage is in relation to future attacks. Moreover, testing and validation of the *Resilient Shield* within an industrial context is left as a future work.

Acknowledgment. This research was supported by the CyReV project (2019-03071) funded by VINNOVA, the Swedish Governmental Agency for Innovation Systems.

REFERENCES

[1] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham *et al.*, "Comprehensive experimental analyses of automotive attack surfaces," in *USENIX Security Symposium*. San Francisco, 2011.

[2] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno *et al.*, "Experimental security analysis of a modern automobile," in *2010 IEEE Symposium on Security and Privacy*, 2010, pp. 447–462.

[3] C. Miller and C. Valasek, "Remote exploitation of an unaltered passenger vehicle," *Black Hat USA*, 2015.

[4] Tencent Keen Security Lab, "Experimental Security Assessment of BMW Cars: A Summary Report," https://keenlab.tencent.com/en/whitepapers / Experimental_Security_Assessment_of_BMW_Cars_by_KeenLab.pdf, 2018, accessed: 2020-09-11.

[5] S. Kamkar, "Drive it like you hacked it: New attacks and tools to wirelessly steal cars," *Presentation at DEFCON*, vol. 23, 2015.

[6] CVE Details, "Security vunerabilities bluelink," https://www.cvedetails.com/vulnerability-list/vendor_id-16402/product_id-37376/Hyundaiusa-Blue-Link.html, accessed: 2020-09-11.

[7] CVE List, "CVE-2019-9493," https : / / cve.mitre.org / cgi - bin/cvename.cgi?name=CVE-2019-9493, accessed: 2020-09-11.

[8] UNECE, "Draft recommendation on cyber security of the task force on cyber security and over-the-air issues of UNECE wp.29 GRVA," UNECE, Tech. Rep., 2018.

[9] "ISO/SAE 21434 Road Vehicles – Cybersecurity Engineering," International Organization for Standardization (ISO), Standard, 2020.

[10] K. Strandberg, T. Olovsson, and E. Jonsson, "Securing the connected car: A security-enhancement methodology," *IEEE Vehicular Technology Magazine*, vol. 13, no. 1, pp. 56–65, 2018.

[11] "Good practices for security of smart cars," ENISA, Tech. Rep., 2019.

[12] "Cyber security and resilience of smart cars," ENISA, Tech. Rep., 2016.

[13] "SAE J3061: Cybersecurity guidebook for cyber-physical vehicle systems," SAE International, Standard, 2016.

[14] EVITA, "EVITA deliverables," https : / / www.evita - project.org / deliverables.html, accessed: 2020-09-11.

[15] M. Islam, A. Lautenbach, C. Sandberg, and T. Olovsson, "A risk assessment framework for automotive embedded systems," *Proceedings of the 2nd ACM International Workshop on Cyber-Physical System Security*, 2016.

[16] Microsoft, "The STRIDE threat model," https://msdn.microsoft.com/en-us/library/ee823878.aspx, 2005, accessed: 2020-09-11.

[17] T. Rosenstatter and T. Olovsson, "Towards a standardized mapping from automotive security levels to security mechanisms," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 1501–1507.

[18] T. Rosenstatter, K. Strandberg, R. Jolak, R. Scandariato, and T. Olovsson, "REMIND: A framework for the resilient design of automotive systems," *IEEE Secure Development*, 2020, in press.

[19] A. Karahasanovic, P. Kleberger, and M. Almgren, "Adapting threat modeling methods for the automotive industry," *15th ESCAR, Berlin*, 2017.

[20] "ISO 26262:2011 Road Vehicles – Functional Safety," International Organization for Standardization (ISO), Standard, 2011.

[21] F. D. Garcia, D. Oswald, T. Kasper, and P. Pavlidès, "Lock it and still lose it—on the (in) security of automotive remote keyless entry systems," in *25th USENIX Security Symposium (USENIX Security 16)*, 2016.

[22] A. Greenberg, "Just a pair of these $11 radio gadgets can steal a car," https://www.wired.com/2017/04/just-pair-11-radio-gadgets-can-steal-car/, accessed: 2020-09-11.

[23] A. Francillon, B. Danev, and S. Capkun, "Relay attacks on passive keyless entry and start systems in modern cars," in *Proceedings of the Network and Distributed System Security Symposium (NDSS)*. ETH Zürich, Department of Computer Science, 2011.

[24] M. L. Psiaki and T. E. Humphreys, "GNSS spoofing and detection," *Proceedings of the IEEE*, vol. 104, no. 6, pp. 1258–1270, 2016.

[25] K. Iehira, H. Inoue, and K. Ishida, "Spoofing attack using bus-off attacks against a specific ECU of the CAN bus," in *15th IEEE Consumer Communications & Networking Conference (CCNC)*, 2018, pp. 1–4.

[26] Q. Meng, L. Hsu, B. Xu, X. Luo, and A. El-Mowafy, "A GPS spoofing generator using an open sourced vector tracking-based receiver," *Sensors*, vol. 19, no. 18, p. 3993, 2019.

[27] CVE List, "CVE-2019-12797," https : / / cve.mitre.org / cgi - bin/cvename.cgi?name=CVE-2019-12797, accessed: 2020-09-11.

[28] Y. Cao, C. Xiao, B. Cyr, Y. Zhou, W. Park *et al.*, "Adversarial sensor attack on LiDAR-based perception in autonomous driving," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 2267–2281.

[29] Cyware Hacker News, "Seven car manufacturers hit by GPS spoofing attacks," https://cyware.com/news/seven-car-manufacturers-hit-by-gps-spoofing-attacks-146701c4, accessed: 2020-09-11.

[30] Help Net Security, "Research shows Tesla Model 3 and Model S are vulnerable to GPS spoofing attacks," https://www.helpnetsecurity.com/2019/06/19/tesla-gps-spoofing-attacks/, accessed: 2020-09-11.

[31] CVE, "CVE-2018-11478," https : / / cve.mitre.org / cgi - bin / cvename.cgi?name=CVE-2018-11478, accessed: 2020-09-11.

[32] D. Schmidt, K. Radke, S. Camtepe, E. Foo, and M. Ren, "A survey and analysis of the GNSS spoofing threat and countermeasures," *ACM Comput. Surv.*, vol. 48, no. 4, May 2016. [Online]. Available: https://doi.org/10.1145/2897166

[33] Pen Test Partners, "Hacking the Mitsubishi Outlander PHEV hybrid," https://www.pentestpartners.com/security-blog/hacking-the-mitsubishi-outlander-phev-hybrid-suv/, accessed: 2020-09-11.

[34] J. Petit, B. Stottelaar, M. Feiri, and F. Kargl, "Remote attacks on automated vehicles sensors: Experiments on camera and lidar," *Black Hat Europe*, vol. 11, p. 2015, 2015.

[35] C. Yan, W. Xu, and J. Liu, "Can you trust autonomous vehicles: Contactless attacks against sensors of self-driving vehicle," *DEF CON*, vol. 24, no. 8, p. 109, 2016.

[36] Tencent Keen Security Lab, "Experimental security assessment on lexus cars," https://keenlab.tencent.com/en/2020/03/30/Tencent-Keen-Security - Lab - Experimental - Security - Assessment - on - Lexus - Cars/, 2020, accessed: 2020-09-15.

[37] P. Murvay and B. Groza, "Practical security exploits of the FlexRay in-vehicle communication protocol," *International Conference on Risks and Security of Internet and Systems*, pp. 172–187, 2019.

[38] Argus Cyber Security, "A remote attack on the Bosch Drivelog connector dongle," https://argus-sec.com/remote-attack-bosch-drivelog-connector-dongle/, accessed: 2020-09-11.

[39] CVE List, "CVE-2016-9337," https : / / cve.mitre.org / cgi - bin/cvename.cgi?name=CVE-2016-9337, accessed: 2020-09-11.

[40] S. Woo, H. J. Jo, and D. H. Lee, "A practical wireless attack on the connected car and security protocol for in-vehicle can," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 993–1006, 2015.

[41] M. Yan, J. Li, and G. Harpak, "Security Research on Mercedes-Benz: From Hardware to Car Control," https://i.blackhat.com/USA-20/Thursday/us-20-Yan-Security-Research-On-Mercedes-Benz-From-Hardware-To-Car-Control.pdf, 2020, accessed: 2020-09-15.

[42] G. H. Ruffo, "Tesla Data Leak: Old Components With Personal Info Find Their Way On eBay," https://insideevs.com/news/419525/tesla-data-leak-personal-info-ebay/, accessed: 2020-09-11.

[43] CVE List, "CVE-2018-11477," https : / / cve.mitre.org / cgi - bin/cvename.cgi?name=CVE-2018-11477, accessed: 2020-09-11.

[44] J. Liu, S. Zhang, W. Sun, and Y. Shi, "In-vehicle network attacks and countermeasures challenges and future directions," *IEEE Network*, 2017.

[45] J. C. Norte, "Hacking industrial vehicles from the internet," http://jcarlosnorte.com/security/2016/03/06/hacking-tachographs-from-the-internets.html, accessed: 2020-09-11.

[46] A. Palanca1, E. Evenchick, F. Maggi, and S. Zanero, "A stealth, selective, link-layer denial-of-service attack against automotive networks," *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, 2017.

[47] P. Murvay and B. Groza, "DoS attacks on controller area networks by fault injections from the software layer," *Proceedings of the 12th International Conference on Availability, Reliability and Security*, 2017.

[48] CVE List, "CVE-2016-2354," https : / / cve.mitre.org / cgi - bin/cvename.cgi?name=CVE-2016-2354, accessed: 2020-09-11.

[49] K. Cho and K. Shin, "Error handling of in-vehicle networks makes them vulnerable," *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016.

[50] J. Dürrwang, J. Braun, M. Rumez, and R. Kriesten, "Security evaluation of an airbag-ECU by reusing threat modeling artefacts," in *2017 International Conference on Computational Science and Computational Intelligence (CSCI)*, 2017, pp. 37–43.

[51] T. Brewster, "BMW updates kills bug in 2.2 million cars that left doors wide open to hackers," https://www.forbes.com/sites/thomasbrewster/2015/02/02/bmw-door-hacking/, 2015, accessed: 2020-09-11.

[52] T. Hunt, "Controlling vehicle features of Nissan LEAFs across the globe via vulnerable APIs," https://www.troyhunt.com/controlling-vehicle-features-of-nissan/, 2016, accessed: 2020-09-11.

[53] M. Wu, H. Zeng, C. Wang, and H. Yu, "INVITED: Safety guard: Runtime enforcement for safety-critical cyber-physical systems," in *54th ACM/EDAC/IEEE Design Automation Conference (DAC)*, 2017, pp. 1–6.

[54] S. Sanwald, L. Kaneti, M. Stöttinger, and M. Böhner, "Secure boot revisited," *17th escar Europe*, 2019.

[55] *MISRA C: Guidelines for the Use of the C Language in Critical Systems 2012*. Motor Industry Research Association, 2013.

[56] T. Karthik, A. Brown, S. Awwad, D. McCoy, R. Bielawski *et al.*, "Uptane: Securing software updates for automobiles," *14th ESCAR Europe*, 2016.

[57] H. Debar, M. Dacier, and A. Wespi, "Towards a taxonomy of intrusion-detection systems," *Computer Networks*, vol. 31, no. 8, pp. 805 – 822, 1999.

[58] C. G. Rieger, D. I. Gertman, and M. A. McQueen, "Resilient control systems: Next generation design research," in *2009 2nd Conference on Human System Interactions*, 2009, pp. 632–636.