Threat Modeling for Automotive Security Analysis

Zhendong Ma and Christoph Schmittner

AIT Austrian Institute of Technology, Digital Safety & Security Department,
Donau-City-Strasse 1, 1220 Vienna, Austria

{Zhendong.Ma, Christoph.Schmittner.fl}@ait.ac.at

Abstract. The technological development of connected and intelligent vehicle creates cybersecurity threats and risks to road safety. Securing automotive systems is one of the biggest challenges for the automotive industry undergoing a profound transformation. As a building block of automotive security, threat modeling is a technique that identifies potential threats in order to find corresponding mitigations. In this paper, we propose a practical and efficient approach to threat modeling for the automotive domain. We extend existing tool support and demonstrate the applicability and feasibility of our approach.

Keywords: security, automotive, threat modeling, safety

1 Introduction

Cars are becoming more and more intelligent and connected. On the flip side, this technological transformation also makes modern vehicles vulnerable to cyberattacks [1, 2, 3]. Cars used to be closed system. The automotive systems were not designed with security in mind. Recent security breaches in the automotive domain raise the issue in the industry and the public, making it clear that security is a critical concern with an impact on public and road safety, especially when new technologies such as autonomous driving and intelligent transport systems (ITS) are becoming reality.

Rigorous security engineering approaches to the development of automotive systems are required to address safety and security of modern vehicles. Security analysis is one of the important building blocks in this process. Threat modeling is a technique for security analysis. As a concept, threat modeling has been extensively covered in many previous works. However, as we observed, there are many misconceptions and confusions on how to apply threat modeling in an efficient and correct way, especially in the emerging field of automotive security. In this paper, we provide a practical guide on conducing threat modeling for automotive system security analysis. Moreover, we propose optimizations to make it more efficient, repeatable and accurate. We also show that our proposal is readily supported by existing tools for practical need in the automotive industry.

In the following, Sec. 2 gives an overview of secure development in the automotive domain. Sec. 3 describes our approach to threat modeling, followed by a proof-of-concept in Sec. 4. Sec. 5 concludes the paper with our plan for future work.

2 Secure Development of Automotive Systems

The automotive industry traditionally has a very high quality and safety standard. As a basis, the automotive industry developed and accepted ISO 26262 [4] as the standard for generic road vehicle functional safety for electrical and electronic (E/E) systems that cover both hardware and software. The development starts with the *concept phase* in which an item is defined followed by activities such as hazard analysis and risk assessment (HARA) and the definition of functional safety concept. An item is a system or an array of systems to implement a function at the vehicle level to which ISO 26262 is applied. HARA identifies safety risks which lead to the definition of safety goals. Automotive safety integrity level (ASIL) is assigned to the safety goals to denote the level of risk reduction to prevent a specific hazard. In the next phase *product development*, the functional safety concept is refined to produce technical safety requirements and hardware and software system are designed, integrated, and tested. Compliance and correctness of the safety goals and their implementation are validated. Safety cases, documentation of all the work products (i.e. all artifacts including information, data, models and source code from the safety activities), are produced as evidence for compliance and certification. Fig. 1 illustrates the overall safety process defined in ISO 26262.

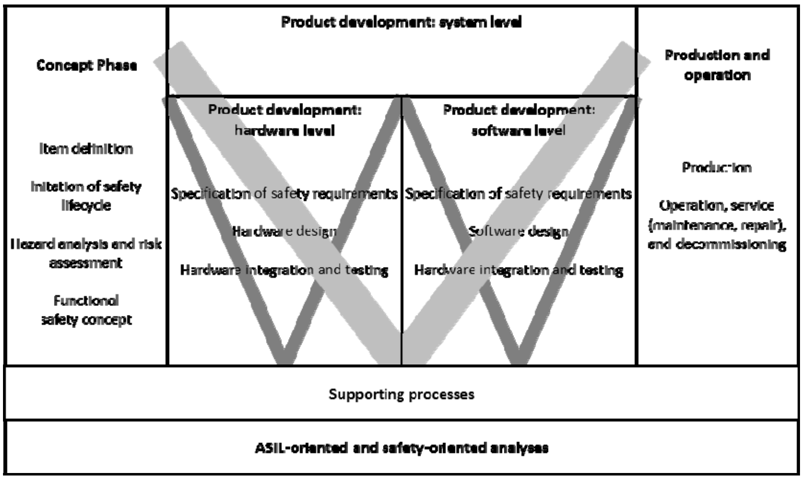

Concept Phase
Item definition
Initation of safety lifecycle
Hazard analysis and risk assessment
Functional safety concept
Product development: system level
Product development: hardware level
Specification of safety requirements
Hardware design
Hardware integration and testing
Product development: software level
Specification of safety requirements
Software design
Hardware integration and testing
Production and operation
Production
Operation, service (maintenance, repair), and decommissioning
Supporting processes
ASIL-oriented and safety-oriented analyses

Fig. 1. The V-model of the safety lifecycle according to ISO 26262

As security becomes an issue for safety in modern vehicles, several attempts have emerged in recent years to tackle secure development of ICT components and systems in the automotive domain. There have been on-going discussions on how to seamlessly integrate security activities into existing safety-oriented automotive development lifecycle [5, 6, 7]. The recent SAE J3061 standard [8] is the most prominent in the industry to define secure development process for cyber-physical vehicle systems. It builds on ISO 26262 and intends to compliment the safety process with security process with interaction points between the two engineering processes. For example, instead of HARA, J3061 defines the activity of Threat Analysis and

Risk Assessment (TARA) to identify potential cybersecurity threats, assess and rate the risk associated with the threats. Being a guideline, J3061 does not require changing the existing development process and is very likely to be adopted by many in the automotive industry to a certain degree in the coming years. Threat modeling is specified in J3061 to identify threats and security risks during design.

In addition to the one-size-fits-all standards, the automotive industry comes up with its own solutions targeting specific part of the development lifecycle. *Macher et al.* [9] proposed to extend HARA with threat modeling STRIDE method for security-aware hazard analysis and risk assessment (SAHARA) to define the ASILs, i.e. add STRIDE-based security analysis as an additional activity to the safety analysis of items defined according to ISO 26262. *Eichler et al.* [10] proposed a modular and flexible approach for security risk assessment in the automotive development process. Activities, tasks, and related work products, roles, and guidance are defined for security analysis. It also includes activities related to threat modeling such as data flow modeling, identification of associated threat and specification of mitigations.

Our focus of the paper is on threat modeling as a best-practice technique for identifying and analyzing security threats and risks in the automotive domain. However, despite the ubiquitous mentioning of threat modeling in various approaches to automotive secure development, mostly threat modeling is defined as an activity required without detailed elaboration on how to do it exactly.

3 Automotive Threat Modeling

Threat modeling *per se* is the activity of defining a theoretical model of perceived threats to a system. The better the assumptions, the closer is the theoretical model to the practical implementation to capture the significant attack vectors [11]. Therefore, threat modeling can be seen as addressing two basic questions:

- How to model a system and its trust assumptions?
- How to model an adversary that captures its motivations, capabilities, and actions including its tactics, techniques, and procedures (TTP)?

3.1 Threat Modeling in Automotive Secure Development Lifecycle

Threat modeling was populated by Microsoft to address software security of web applications in requirement and design phase. Although automotive systems share many commonalities with standard IT systems, there are also differences which require domain specific techniques and considerations.

Fig. 2 shows the conceptual view of a systematic approach to applying threat modeling technique to automotive security analysis. Solid and dotted arrows indicate information flows. Threat modeling as an activity should be performed in all phases (concept, product development, and production and operation) of the development lifecycle. Although the basic technique remains the same, threat modeling will have different input with respect to the details of the system as it evolves along the development lifecycle. Moreover, threat modeling will have different objectives in each phase. In the concept phase, threat modeling is based on system concept and high-level system design with less technical details. The outcome of the threat modeling is the high-level security requirements and security concept. In the product development phase, the input to the threat modeling will be system design specifications as well as implementation details. The objective of threat modeling is to define technical security requirements for functional and security design, discover design vulnerability and flaws, and specify comprehensive security requirements that can be verified and validated in unit and integration testing along the V model. It is very likely that threat modeling will be an iterative process due to the continuous development and modification of system design and implementation details. In the production and operation phase, threat modeling serves as a preparation for conducting actual penetration testing on finished automotive components and systems. It identifies high-risk inputs and keeps a checklist of things to audit which helps to prioritize entry points that could yield the most return during a pentest [12]. Since threat modeling includes the definition of not only threats but also mitigations, outcomes from threat modeling might have significant impact and modification to the design and implementation of the automotive systems in the development lifecycle.

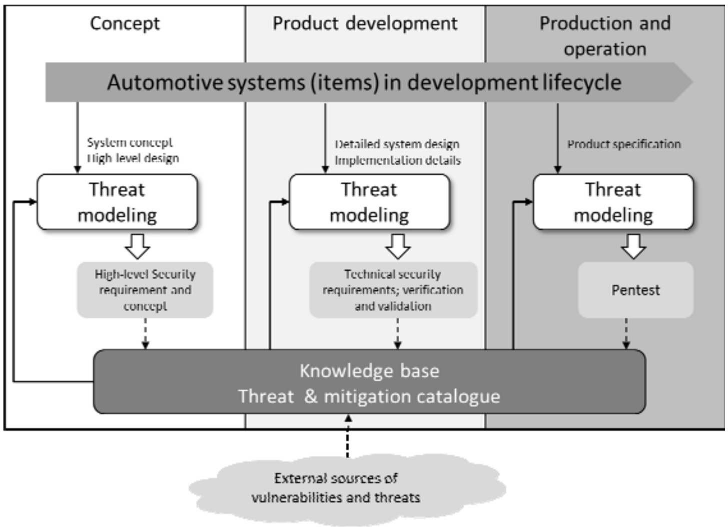

Concept
Product development
Production and operation
Automotive systems (items) in development lifecycle
System concept
High level design
Detailed system design
Implementation details
Product specification
Threat modeling
Threat modeling
Threat modeling
High-level Security requirement and concept
Technical security requirements; verification and validation
Pentest
Knowledge base
Threat & mitigation catalogue
External sources of vulnerabilities and threats

Fig. 2. Overview of systematic threat modeling in automotive secure development lifecycle

The knowledge base on the bottom of Fig. 2 illustrated a central store of information on threats and the corresponding mitigations, which is used in threat modeling in different phases concerning various system models. The knowledge base should be continuously enriched by the output from threat modeling activities with additional threats and mitigations, enabling the reuse of threat modeling artefacts throughout different projects. Further, related vulnerabilities and threats from external sources such as vulnerability databases, hacker communities, and security researchers should be timely incorporated into the threat and mitigation catalogue. The dotted arrows indicate that ingress information to the knowledge base which requires processing to match the format and semantics of the threat and mitigation catalogue.

3.2 Threat Modeling of Automotive Systems

The main focus of threat modeling is software. Generally it includes:

1. Model a system by drawing the system architecture in Data-flow Diagram (DFD), adding system details to the elements in the DFD, and draw the trust boundaries.
2. Identify threats stemmed from data flows by using a threat identification methodology such as STRIDE. An assessment of the severity of the threats can be added.
3. Address each threat by redesigning the system, adding mitigation, or ignoring it if the risk is acceptable.
4. Validate the threat modeling diagram against actual system and all identified threats are addressed.

Conventionally, there are five types of elements in a DFD diagram: process, data store, data low, external interactor, and trust boundary. A process can be any software component that takes input and performs actions and/or generates output. Processes can have different levels of granularity. A high-level process can be decomposed into low-level processes in a hierarchical way. For example, a Level 0 process “Head Unit” can be decomposed into Level 1 processes of “Communication Gateway”, “Linux OS”, “Applications”, and “HMI” etc. Depending on the available system details and threat identification needs, a process can be further decomposed into lower-level components such as specific Linux kernel modules. Example data stores can be firmware, filesystem, or memory. A data flow represents the flow of data between elements. For example, a data flow can be a protocol specific communication link such as CAN Bus, FlexRay, or HTTPs. An external interactor is either a human user or a user agent that interacts with a process from the outside. Trust boundaries divide the elements in the diagram into different trust zones, e.g. elements reside in the in-car systems and external hosts communicated from untrusted open networks.

When identifying threats, different methodologies can be applied. The Spoofing, Tampering, Repudiation, Information disclosure, Denial of Service, and Elevation of privilege (STRDE) methodology is one of the most popular ones partially due its easy-for-developer origin [13] and extensive documentation of applications [14]. However, depending on the granularity of the system information available and the timing of the threat modeling in the development lifecycle, alternative methodologies can also be used for optimal cost-benefit results. For example, sometimes enumerating potential attacks on each of the elements in a brainstorming session will be sufficient for improving the security posture of the design.

Mitigations are the opposite part of the threats. Mitigations can be technical or organizational. The linking of mitigations to the threats ensures that all identified threats will be considered and addressed. Moreover, it also puts mitigations into perspective with the overall security architecture as well as other requirements such as usability, safety, and budget constraints for making sound design decisions.

Validating theoretical models against actual systems will ensure the correctness of the results from the threat modeling. Validating all identified threats are addressed provides additional layer of quality control on the security process.

3.3 Knowledge Base

An important prerequisite for generating meaningful and correct results from threat modeling is the understanding of both the domain-specific system and the threat landscape. An understanding of the system can be gained from domain experts within the engineering team. However, a comprehensive understanding of the threat landscape applicable to threat modeling is a challenging task due to the heterogeneity, complexity, and interdependency of modern computer systems and the dynamics of changing threat landscape. A significant amount of experience and knowledge is required to correctly and efficiently identify and predict known and even unknown threats once the system is in the wild. Besides, the results should be consistent across all threat modeling sessions and human error and oversight should be minimized as much as possible.

Although human expertise will always play a main role in this process, the establishment and maintenance of a knowledge base (cf. Fig. 2) in which threats and mitigations are collected, categorized, and updated that are applicable to the context of different system diagrams will be a viable way to increase efficiency and reduce cost and human errors. In such a way, complex system can be analyzed semi-automatically by leveraging previous results; repeated work can be kept at minimal. This also allows reusing analysis efforts for future projects and even across domains. Knowledge databases for web security can be used as an example for considering threats to the backend and web-communication parts of an automotive update system. As we will show in the next section, current tool is able to support such a vision.

4 Implementation

The existence of easy-to-use tools makes it relatively straightforward to apply threat modeling to many systems. In the past years, Microsoft has developed a tool called Threat Modeling Tool (TMT) [15]. By default, it targets web application development using STRIDE method. However, the latest release in 2016 also provides possibilities to create new threat templates, which enables us to extend TMT so that it is suitable for threat modeling for automotive systems.

When creating a new template, the most important parts are *stencils* for drawing DFD diagram and *threat types* that define threat and mitigation catalogues. We create stencils for automotive components such as Electronic Control Unit (ECU), in which we add additional details to describe the component. Once defined, these additional details provide rich information about an automotive component during threat modeling. The threat catalogue can be defined by threat properties in the TMT template. Each threat type includes title, threat description. More importantly, it has fields of *include* and *exclude*, which can be used for writing simple logical expressions such as `source is [stencil name]` so that threats can be automatically generated on an element of a DFD diagram when the condition is satisfied. Threat properties are grouped by threat types. For example, in the default TMT template, the threat types are defined as spoofing, tampering, repudiation, information disclosure, denial of service, and elevation of privilege. Similar to stencils, threat properties can be flexibly extended for addition fields. Accordingly, specific mitigations can be defined in the corresponding threat properties.

Fig. 3 and Fig. 4 show an example of the proof-of-concept implementation of our approach to threat modeling for automotive security analysis. The example is based on our experience of conducting TARA of an automotive cockpit unit that equipped with wireless communication modules for various remote access functionalities including maintenance and over-the-air (OTA) software update [16]. Fig. 3 shows the top-level DFD diagram of the system. In the center of the figure is the Operator controller which is an ARM-based System-on-Chip (SoC) microcontroller running on embedded Linux. The Human-machine Interface (HMI) enables an operator in the cockpit to issue command and monitor the status of the vehicle. Because of the wireless module, it can communicate with update severs at the back-end. Operators and engineers can access the controller remotely through a VNC client. The controller has a firmware data store within the physical boundary of the vehicle and it also connects to ECUs through the CAN bus interface. The panel on the right shows some customized stencils with system-specific details.

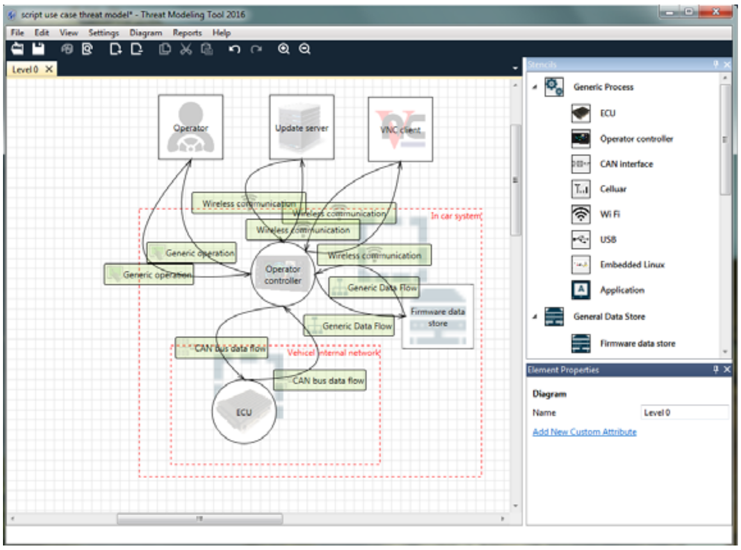
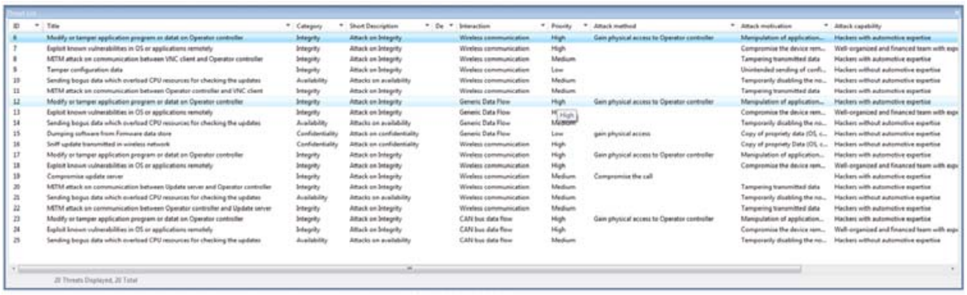


Fig. 3. Example of top level DFD diagram of automotive unit

Fig. 4 shows the automatically generated threats based on the DFD diagram. It shows some selected fields of the threat list such as Title, Short Description, Attack method, Attack motivation, and Attack capability. Since we can define additional fields in the template, there is no limitation what information to be included in the threat description. Due to space constraints, mitigations are not shown here. Note that in this example, we conduct the threat modeling based on a generic CIA method, i.e.

we enumerate and identify the attacks on confidentiality, integrity, and available. The reason is due to the lack of detailed technical details in the concept phase. It also shows that by extending the TMT templates, one can flexible choose analysis method to best suit specific need and level of abstraction.



ID	Title	Category	Short Description	De	Interaction	Priority	Attack method	Attack motivation	Attack capability
6	Modify or tamper application program or datat on Operator controller	Integrity	Attack on Integrity		Wireless communication	High	Gain physical access to Operator controller	Manipulation of application...	Hackers with automotive expertise
7	Exploit known vulnerabilities in OS or applications remotely	Integrity	Attack on Integrity		Wireless communication	High		Compromise the device rem...	Well-organized and financed team with exp
8	MITM attack on communication between VNC client and Operator controller	Integrity	Attack on Integrity		Wireless communication	Medium		Tampering transmitted data	Hackers with automotive expertise
9	Tamper configuration data	Integrity	Attack on Integrity		Wireless communication	Low		Unintended sending of confi...	Hackers without automotive expertise
10	Sending bogus data which overload CPU resources for checking the updates	Availability	Attacks on availability		Wireless communication	Medium		Temporarily disabling the no...	Hackers without automotive expertise
11	MITM attack on communication between Operator controller and VNC client	Integrity	Attack on Integrity		Wireless communication	Medium		Tampering transmitted data	Hackers with automotive expertise
12	Modify or tamper application program or datat on Operator controller	Integrity	Attack on Integrity		Generic Data Flow	High	Gain physical access to Operator controller	Manipulation of application...	Hackers with automotive expertise
13	Exploit known vulnerabilities in OS or applications remotely	Integrity	Attack on Integrity		Generic Data Flow	High		Compromise the device rem...	Well-organized and financed team with exp
14	Sending bogus data which overload CPU resources for checking the updates	Availability	Attacks on availability		Generic Data Flow	Medium		Temporarily disabling the no...	Hackers without automotive expertise
15	Dumping software from Firmware data store	Confidentiality	Attack on confidentiality		Generic Data Flow	Low	gain physical access	Copy of propriety data (OS, c...	Hackers without automotive expertise
16	Sniff update transmitted in wireless network	Confidentiality	Attack on confidentiality		Wireless communication	High		Copy of propriety Data (OS, c...	Hackers without automotive expertise
17	Modify or tamper application program or datat on Operator controller	Integrity	Attack on Integrity		Wireless communication	High	Gain physical access to Operator controller	Manipulation of application...	Hackers with automotive expertise
18	Exploit known vulnerabilities in OS or applications remotely	Integrity	Attack on Integrity		Wireless communication	High		Compromise the device rem...	Well-organized and financed team with exp
19	Compromise update server	Integrity	Attack on Integrity		Wireless communication	Medium	Compromise the call		Hackers with automotive expertise
20	MITM attack on communication between Update server and Operator controller	Integrity	Attack on Integrity		Wireless communication	Medium		Tampering transmitted data	Hackers with automotive expertise
21	Sending bogus data which overload CPU resources for checking the updates	Availability	Attacks on availability		Wireless communication	Medium		Temporarily disabling the no...	Hackers without automotive expertise
22	MITM attack on communication between Operator controller and Update server	Integrity	Attack on Integrity		Wireless communication	Medium		Tampering transmitted data	Hackers with automotive expertise
23	Modify or tamper application program or datat on Operator controller	Integrity	Attack on Integrity		CAN bus data flow	High	Gain physical access to Operator controller	Manipulation of application...	Hackers with automotive expertise
24	Exploit known vulnerabilities in OS or applications remotely	Integrity	Attack on Integrity		CAN bus data flow	High		Compromise the device rem...	Well-organized and financed team with exp
25	Sending bogus data which overload CPU resources for checking the updates	Availability	Attacks on availability		CAN bus data flow	Medium		Temporarily disabling the no...	Hackers without automotive expertise

20 Threats Displayed, 20 Total

Fig. 4. Automatically generated threats

The advantage is obvious: by maintaining a continuously updated threat catalogue, the process of threat modeling becomes more efficient and the result more accurate due to the accumulated knowledge of the threat landscape. If a threat other than the ones in the knowledge base is identified on a new system, it can be added to the list of automatically generated threat list and be used in the next time.

5 Conclusion

Security is one of the biggest challenges to connected and intelligent vehicle. Threat modeling is an effective technique to identify threats and mitigations during security analysis of automotive systems. We demonstrated that threat modeling, using existing tools, can be a useful and efficient analysis method for automotive security in different phases in the automotive development lifecycle.

In the next steps, we will further validate our approach and tool in industry-related projects. We will investigate how to import existing threat and mitigation catalogues to extend the knowledge base in the tool. We will also integrate threat modeling into software framework supporting automotive system development lifecycle that considers both safety and security. Consistency between the models for analysis and design during development is an important issue in model-based engineering [17]. We will investigate how to connect the models for security analysis (e.g. DFD) with the models for system engineering (e.g. SysML).

References

1. Checkoway, S., McCoy, D., Kantor, B., Anderson, D., Shacham, H., Savage, S., Koscher, K., Czeskis, A., Roesner, F., Kohno, T.: Comprehensive Experimental Analyses of Automotive Attack Surfaces. In: Proceedings of the 20th USENIX Conference on Security (2011)

2. C. Miller and C. Valasek: Remote Exploitation of an Unaltered Passenger Vehicle. Technical Report (2015)
3. Keen Security Lab of Tencent: Car Hacking Research: Remote Attack Tesla Motors. Keen security lab blog (2016)
4. International Organization for Standardization: ISO 26262 Road vehicles - Functional safety (2011)
5. E. Schoitsch, C. Schmittner, Z. Ma, and T. Gruber: The Need for Safety & Cyber-Security Co-engineering and Standardization for Highly Automated Automotive Vehicles. 19th International Forum on Advanced Microsystems for Automotive Applications (AMAA 2015), Berlin, Germany (2015)
6. C. Schmittner and Z. Ma: Towards a Framework for Alignment Between Automotive Safety and Security Standards. SAFECOMP Workshops 2015: 133-143 (2015)
7. Information-technology Promotion Agency (IPA) Japan: Approaches for vehicle information security (2013)
8. SAE International: J3061 Cybersecurity Guidebook for Cyber-Physical Vehicle Systems (2016)
9. G. Macher, H. Sporer, R. Berlach, E. Armengaud and C. Kreiner: SAHARA: A security-aware hazard and risk analysis method. 2015 Design, Automation & Test in Europe Conference & Exhibition (DATE), Grenoble (2015)
10. J. Eichler and D. Angermeier: Modular risk assessment for the development of secure automotive systems. Conference: 31. VDI/VW-Gemeinschaftstagung Automotive Security, At Wolfsburg, Volume: VDI-Berichte 2263 (2015)
11. Josiah Dykstra: Essential Cybersecurity Science - Build, Test, and Evaluate Secure Systems. O'Reilly (2015)
12. Craig Smith: The car hacker's handbook - a guide for the penetration tester. No Starch Press (2016)
13. Adam Shostack: Experiences threat modeling at Microsoft. Modeling Security Workshop. Dept. of Computing, Lancaster University, UK (2008)
14. Adam Shostack: Threat Modeling: Designing for Security. John Wiley & Sons (2014)
15. Microsoft Threat Modeling Tool 2016 https://www.microsoft.com/en-us/download/details.aspx?id=49168
16. C. Schmittner, Z. Ma, C. Reyes, O. Dillinger and P. Puschner: Using SAE J3061 for Automotive Security Requirement Engineering. SAFECOMP Workshops (2016)
17. A. Joshi, M. P. Heimdahl, S. P. Miller, and M. W. Whalen: Model-Based Safety Analysis. NASA, CR-2006-213953 (2006)