RESOLUTION OF SAFETY RELEVANT SECURITY THREATS IN THE SYSTEM ARCHITECTURE DESIGN PHASE ON THE EXAMPLE OF AUTOMOTIVE INDUSTRY

Japs, Sergej;
Anacker, Harald

Fraunhofer Research Institute for Mechatronic Systems Design IEM

ABSTRACT

Cyber-physical systems (CPS), like autonomous vehicles, are intelligent and networked. The development of such systems and its components requires interdisciplinary cooperation between different stakeholders. A lack of system understanding between stakeholders can lead to unidentified and unresolved security threats & safety hazards in early engineering phases, resulting in high costs in product development and potentially compromises compliance with the safety of CPS.
Model-based systems engineering (MBSE) improves the system understanding between stakeholders by using models.
However, MBSE approaches only partially address security threats & safety hazards. In particular, their integrative consideration is not taken into account.
Established security & safety approaches are either only applicable to specific disciplines or only partially consider security threats & safety hazards.
In the context of this paper we present a method for the resolution of safety relevant security threats in the system architecture design phase using design patterns.
We illustrate our approach with the example of the automotive sector.
Finally, we present an evaluation of the method, based on an 8 week project with 67 master students.

Keywords: Systems Engineering (SE), Design for X (DfX), Early design phases, Risk management, Security & Safety

Contact:
Japs, Sergej
Fraunhofer Research Institute for Mechatronic Systems Design IEM
Product Engineering
Germany
sergej.japs@iem.fraunhofer.de

Cite this article: Japs, S., Anacker, H. (2021) 'Resolution of Safety Relevant Security Threats in the System Architecture Design Phase on the Example of Automotive Industry', in *Proceedings of the International Conference on Engineering Design (ICED21),* Gothenburg, Sweden, 16-20 August 2021. DOI:10.1017/pds.2021.517

1 INTRODUCTION

Cyber-physical systems (CPS), like autonomous vehicles, are intelligent and networked. The development of such systems and its components is characterized by the close cooperation of mechanical, electrical and software engineering (Gausemeier et al., 2014). The interdisciplinarity and complexity of these systems leads to an increasing challenge for an effective and efficient development. A lack of system understanding between the stakeholders can lead to unidentified security threats & safety hazards in the system architecture design phase. This can result in high costs in product development.

For distinction we introduce the following definitions: A security threat exists in a specific item (component, function, requirement etc.) when hackers can exploit a vulnerability to gain unauthorized access to the system. A safety hazard in a specific item exists if the system can cause physical damage to the system or its environment. A security hazard is given, if both cases are fulfilled. In the following we will focus on security hazards.

A lack of security in a CPS can affect the safety of the CPS (security hazard) and damage the corporate reputation. In 2015 hackers demonstrated an attack on a moving SUV (Greenberg, 2015). In this case, the infotainment system was compromised by a remote hack, allowing the hackers to take control of the vehicle. This triggered a product recall of 1.4 million vehicles for the affected company (Goldman, 2015).

In order to resolve security hazards in engineering at an early stage, a holistic view of the system to be designed is necessary. This includes the involvement of several stakeholders from different disciplines, most of them not familiar with security. Model-based systems engineering (MBSE) improves the system understanding between stakeholders by using models. SysML is the de facto modeling language in MBSE (Dori, 2016). The use of model-based design patterns, assists in the resolution of security hazards by reusing existing solutions to design problems with security hazards.

Based on the literature we have analyzed, we formulate the following research question: *What methodical steps are necessary to support an interdisciplinary team of stakeholders in a workshop so that they can jointly identify and resolve security hazards, and how must SysML constructs be designed for this purpose so that they can be applied in the workshop?*

We have selected the literature on the following criteria: Number of citations, publications preferably from the last 5 years. Established security- & safety approaches are partially only applicable to specific engineering disciplines, like software engineering (Howard and Lipner, 2016; Mead et al., 2005; Stølen et al., 2002; Fernandez-Buglioni, 2013; Heisel et al., 2019). Other approaches are applicable interdisciplinary on system level, but only partially consider security hazards. The SREP approach does not consider safety (Rehman et al., 2018), while the following approaches do not consider security Anacker et al. (2020); ISO (2018a, 2015); Pohl (2016); Rupp et al. (2014). The cybersecurity guidebook for cyber-physical vehicle systems (SAE, 2016) and the SAHARA approach (Macher et al., 2015) consider security hazards. Unfortunatelly the cybersecurity guidebook and SAHARA do not use models. Approaches like Cheng et al. (2019); Amorim et al. (2017) use models, but do not use SysML. Approaches like security by MBRE (Japs, 2020a) or SAVE (Japs, 2020b) use SysML within the context of MBSE. However, a concrete method for the resolution of security hazards is missing. The following approaches are suitable for use in workshops and support the identification of new security vulnerabilities: (Japs et al., 2020) supports stakeholders in visualizing threat cases by using a 3D environment, while (Tekaat et al., 2019) extends Design Thinking to consider security. However, both approaches do not support the resolution of identified security vulnerabilities. Furthermore we have analyzed several sources in which security or safety design patterns were presented (Anacker et al., 2020; Amorim et al., 2017; Cheng et al., 2019; Fernandez-Buglioni, 2013; Heisel et al., 2019). Most of the design patterns were unsuitable in terms of structure, description and presentation for application in workshops with an interdisciplinary team of stakeholders, for the following reasons: Most of the design patterns were intended for application by IT experts. This was reflected in the very high level of detail for IT systems and also in the use of UML instead of SysML. On the other hand, design patterns that were tailored for use in MBSE did not consider the security aspect.

In Section 2 we present the required foundation for our approach. In Section 4 we introduce our RE-EDIT method, using an example from the automotive domain (cf. Section 3). The name RE-EDIT is composed of letters from the title. In addition, the name is intended to express that the application of the method results in a re-edit of the initially created system architecture. This is done first by identifying

https://doi.org/10.1017/pds.2021.517 Published online by Cambridge University Press

security hazards in the system architecture. Subsequently, the security hazards are resolved by applying design patterns which result in a change to the system architecture. In Section 5 we present the results of the evaluation of our method based on an 8 week project with 67 interdisciplinary master students. Finally we summarize the paper and give an outlook for future work.

2 FUNDAMENTALS

In this section, we present the underlying approaches for RE-EDIT. In Section 4, we illustrate the use of these approaches in conjunction with our method. As the fundamental approach we use the **CONSENS** method (Gausemeier et al., 2014). CONSENS is an MBSE approach and designed for the interdisciplinary design of complex intelligent technical systems. The method defines, centrally for an MBSE approach, based on an own modeling language the design of a system model in the concept phase. The CONSENS method has been applied in numerous industrial projects in the product conception phase, e.g. to develop smart home products (Unity, 2020) or as a general method for designing mechatronic products (SmartMechatronics, 2020).

The CONSENS method requires the design of different partial models for the different aspects of a technical system. Application Scenarios or User Storys describe a situation-specific view of the described system. In the System Structure, the components of the system and the relationships between the components are modelled as a white box. In the partial model Behavior, the system behavior is modeled by activities, states or sequences. By means of cross-relationships, the different partial models can be connected with each other. Relationships within these models are distinguished according to the following relationship types: information, energy, substance and if unclear logical relationship.

While CONSENS defines its own graphical modeling language, we use the modeling language **SysML** (OMG, 2015), as the de facto modeling language in systems engineering (Dori, 2016). This defines a requirements diagram and several structure and behavior diagrams. In order to be able to use the CONSENS method in combination with the modeling language SysML, we use a corresponding **SysML4CONSENS** profile (Dumitrescu, 2013). Profiles allow the modelling language to be extended by adding further stereotypes. E.g. the stereotype "information" (cf. Figure 1 [TC-IBD]), allows the relationship "Sensor data" between the environmental element "Hacker" and the system component "Multi purpose camera" to be categorized as an information relationship. For better visualisation of the elements with different stereotypes, we use a color scheme based on the CONSENS modelling language. According to the SysML standard, color schemes are not relevant for semantics and can be chosen arbitrarily. By using stereotypes in conjunction with a color scheme, the semantic of colored model elements is given.

Appropriate risk classification schemes are necessary to prioritize engineering activites in MBSE. To prioritize engineering activities with respect to safety, we use the **ASIL** (Automotive Safety Integrity Level) risk classification scheme defined by ISO 26262 (ISO, 2018a). To determine the ASIL values, the following criteria must be evaluated in different grades: Severity, exposure and controllability. Based on this, the ASIL values can be determined with the help of a mapping table. In our approach, we use the ASIL risk classification scheme to evaluate system functions.

To prioritize engineering activities with respect to security, we use the often quoted **SAHARA** risk classification scheme (Macher et al., 2015). To determine the Security Levels, the following criteria must be evaluated in different grades: Required resources, required know-how and threat level. Based on this, the Security Level values can be determined with the help of a mapping table. In our approach, we use the SAHARA risk classification scheme to evaluate system functions. SAHARA can be used to determine safety-relevant security system functions in particular. This is defined in SAHARA as follows: If a security item has a criticality level greater than 2, it is derived as a further safety item, with a potential security hazard.

We use Microsoft's **STRIDE** model (Shostack, 2014) to classify security threats. It consists of the following categories: Spoofing (unauthorized access), Tampering (malicious modification of data), Repudiation (non-recognition of an action), Information Disclosure (publication/access of secret information), Denial of Service (denial of access to a service) and Evaluation of Privilege (extension of rights).

https://doi.org/10.1017/pds.2021.517 Published online by Cambridge University Press

3 BACKGROUND & APPLICATION EXAMPLE

The SecForCARs research project (SecForCARs, 2020) consists of 14 partners from industry and research, among them Audi, Bosch, Infineon, different security companies and universities. The consortium is working on solutions for the consideration of security & safety for future autonomous vehicles along the V-model. As part of the consortium we develop methods like RE-EDIT for the system design phase. We will use platooning as our application example. Platooning describes the networking of several vehicles which, with the help of a technical control system, can drive one behind the other at very short distances without impairing traffic safety. In SecForCARs platooning was identified as the most important use case for autonomous cars. Especially the manipulation of sensor data was critically assessed by the project partners.

We use the following platooning situation: The vehicle platoon reaches an intersection. The platoon leader recognizes an obstacle. A hacker could be sitting in the cafe. The vehicle of the platoon leader can brake, evade, do nothing and warn the platoon participants. Different user stories can belong to the described situation. In MBSE user stories represent a general form of system requirements. In context of this paper we illustrate RE-EDIT on the following user story: (Warn Driver) "*The driver's steering wheel vibrates if an obstacle is detected by the platoon leader's vehicle*".

4 METHOD

In this section we present our RE-EDIT method, which is designed for use in workshops with a team of interdisciplinary stakeholders. Our method supports the stakeholders in resolving identified security hazards in a high level system model by using design patterns.

RE-EDIT contains the following phases: 1 Identification of security hazards in the system model using our previous method SAVE. (Japs, 2020b). 2 Selection of appropriate security hazard design patterns. 3 Resolution of security hazards in the system model using security hazard design patterns.

4.1 Identification of security hazards in the system model

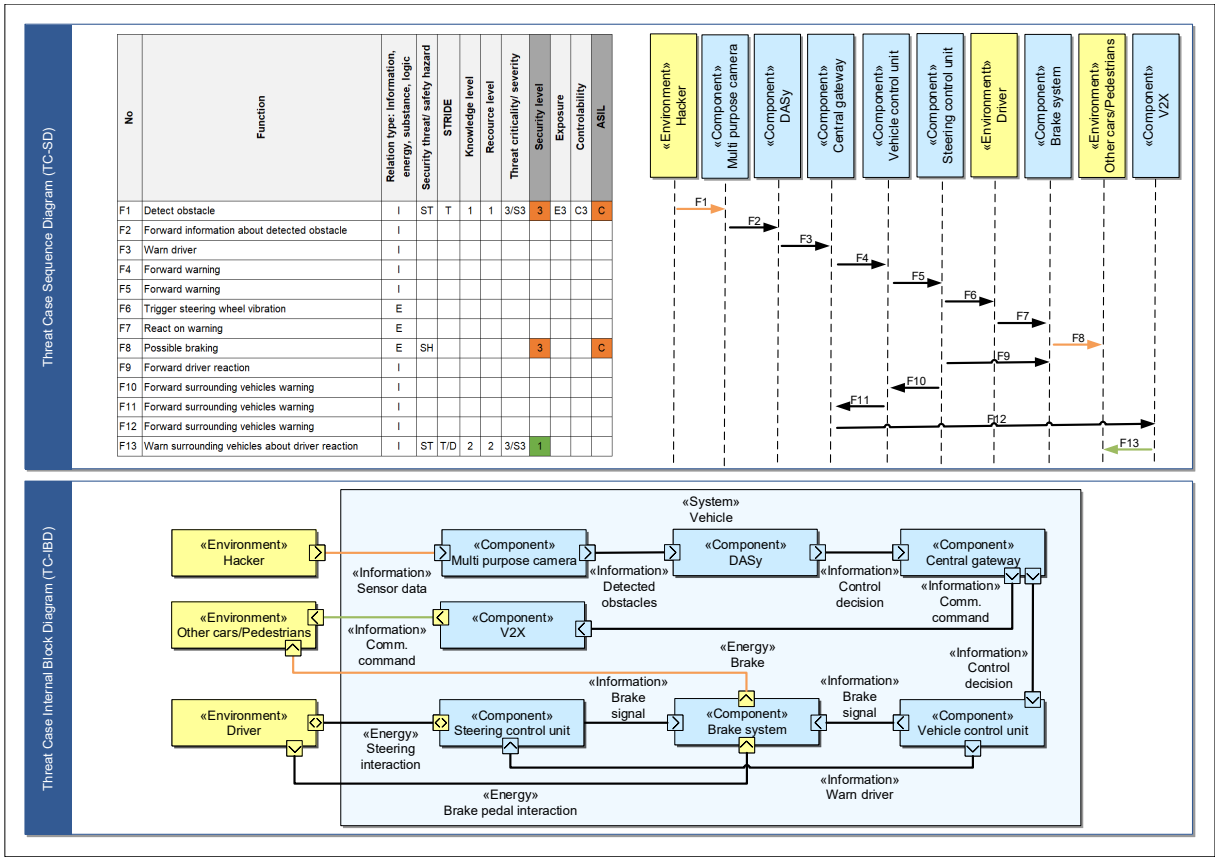
In this section we explain how security hazards can be identified in the system model from a white box perspective. To ensure realistic modelling, our example is based on components and architectural conditions from Bosch (cf. Japs (2020b) for a cheat sheet). In the industrial environment, experts and internal documents replace such cheat sheets.

In Figure 1 we show two partial models of the system model. The first partial model is a SysML sequence diagram representing a threat case (TC-SD). This represents a sequence of interaction between individual system and environmental elements. Here, the interaction takes place through the execution of functions. For the determination of security hazards, each function is given a set of attributes, as an aid for discussions in the workshop: CONSENS relationship type, security/safety relevance, STRIDE category, security level, ASIL. If a function also appears in another TC-SD, the attribute values determined are reused. Compared to classical security approaches, TC-SD correspond in principle to attack trees. In attack trees a diagram shows the sequence in which a target system can be attacked.

The shown TC-SD refines the user story "Warn Driver" (cf. Section 3), which has weaknesses at various points. A hacker can trigger the process by manipulating sensor data (F1-F2), potentially causing physical damage (F7-F8). We evaluate the F1 function as follows: We focus on the detection of obstacles in terms of information processing (CONSENS stereotype: information). The detection in F1 can be manipulated, but no direct physical damage results from this manipulation. Thus, it is a possible security Threat. In accordance with the STRIDE model, this type of manipulation corresponds to the Tampering category. In comparison with SAHARA, no special knowledge (value: 1) and resources (value: 1) are required to detect false obstacles, so that a hazardous situation can be caused with little effort (value: 3/S3). This results in a high security level (level: 3). For example, a teenager could put a cardboard bag on the street to cause a response to detected obstacles. Due to the high criticality, the additional evaluation according to ASIL is carried out according to SAHARA. In comparison with ASIL, the probability (value: E3) for such a case is high and the quick reaction by the driver (value: C3) is limited. This results in the high ASIL value 3. Thus F1 and according to this procedure also F8 (braking) must be considered prioritized in the workshop. Functions F1 and F13 were assigned to the STRIDE security threat class Tampering (T), since both sensor data and radio traffic can be manipulated in terms of content. The F13 function has additionally been assigned to the Denial of Service (D) security threat class because radio

https://doi.org/10.1017/pds.2021.517 Published online by Cambridge University Press

communication can be maliciously interrupted e.g. by a jammer. Although F13 (warn other vehicles) appears critical, it does not have to be prioritized in the workshop due to its low rating.



No	Function	Relation type: Information, energy, substance, logic	Security threat/ safety hazard	STRIDE	Knowledge level	Recource level	Threat criticality/ severity	Security level	Exposure	Controllability	ASIL
F1	Detect obstacle	I	ST	T	1	1	3/S3	3	E3	C3	C
F2	Forward information about detected obstacle	I									
F3	Warn driver	I									
F4	Forward warning	I									
F5	Forward warning	I									
F6	Trigger steering wheel vibration	E									
F7	React on warning	E									
F8	Possible braking	E	SH					3			C
F9	Forward driver reaction	I									
F10	Forward surrounding vehicles warning	I									
F11	Forward surrounding vehicles warning	I									
F12	Forward surrounding vehicles warning	I									
F13	Warn surrounding vehicles about driver reaction	I	ST	T/D	2	2	3/S3	1			

Figure 1. Identification of security hazards in the system model for the user story "Warn Driver"

The second partial model represents a system architecture in the form of a SysML Internal Block Diagram, which is formed from the set of TC-SDs created in the workshop. We call it TC-IBD (Threat Case Internal Block Diagram). For better differentiation of modeling elements and for better visualization in the workshop, we use the stereotypes and color scheme from CONSENS. The TC-IBD serves as a communication tool in the workshop for locating and discussing security-critical interfaces between individual system components. With the help of the security & safety level of the individual functions from the TC-SDs, critical component relationships in the system architecture can be highlighted for further discussion and decision-making. For example, due to the high ASIL value of F8 (braking) in the TC-SD, the relationship between the system element "Brake system" and the environmental element "Other cars/Pedestrians" is particularly highlighted in the TC-IBD.

4.2 Template for workshop applicable security hazard design patterns

Based on the identified design problems, suitable design patterns are selected from a catalog and applied. A design pattern must be described by attributes such as name, application context, problem description, etc. Such attributes support the selection of suitable design patterns. In order not to hinder the creativity process in the workshop, a quick selection of solution patterns must be ensured, by meeting the following requirements: (R1) Design patterns must be described only by the most important attributes, while the text must not contain unnecessary details. (R2) The idea of the design pattern must be understood immediately. No complicated model constructs must be used. Easy-to-understand examples with easy-to-understand model elements support this. (R3) Different model elements need to be identified for quick visual recognition. Color schemes in conjunction with stereotypes provide support here. The approaches we analyzed, which use solution pattern catalogs (Anacker et al., 2020; Amorim et al., 2017; Cheng et al., 2019; Fernandez-Buglioni, 2013; Heisel et al., 2019), do not fully meet these requirements. In particular, none of these approaches met R2 and R3. To satisfy R1-R3, we propose the

https://doi.org/10.1017/pds.2021.517 Published online by Cambridge University Press

following template, which we illustrate based on the DIDS design pattern. The idea of the DIDS design pattern is to use an attack database that the system accesses (cf. Figure 2).

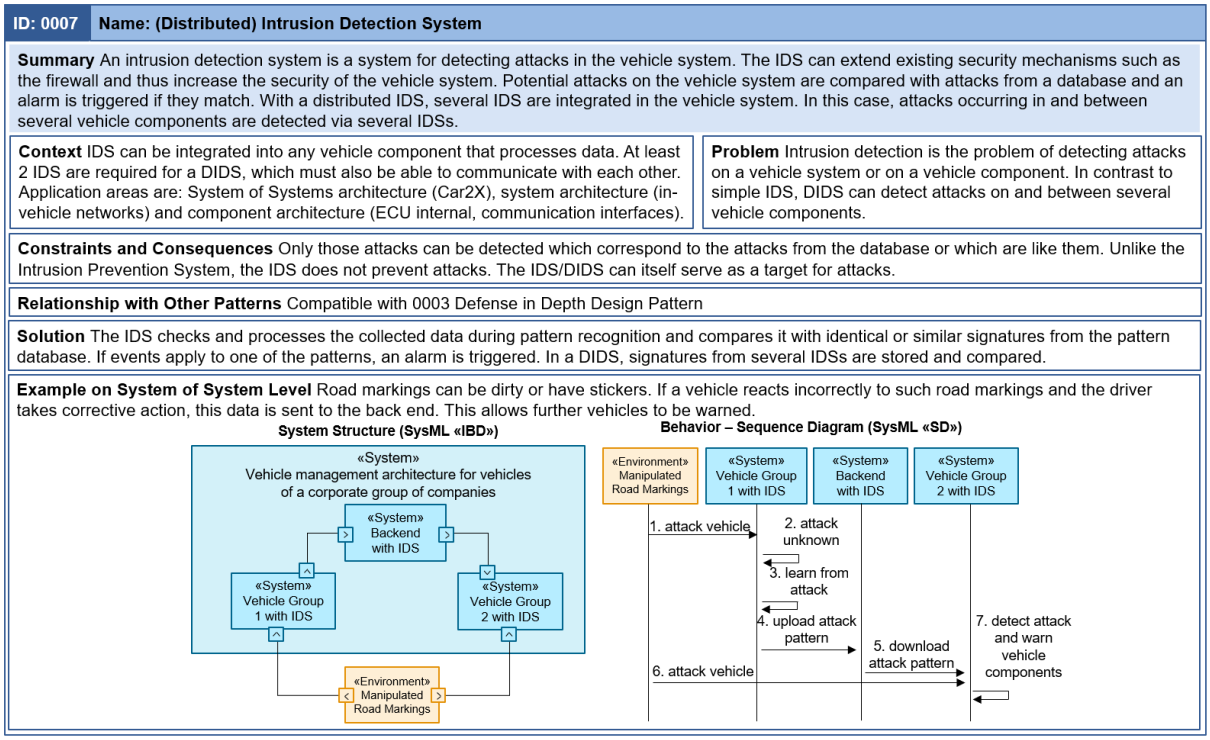


Figure 2. (Distributed) intrusion detection system design pattern

4.3 Resolution of security hazards in the system model using design patterns

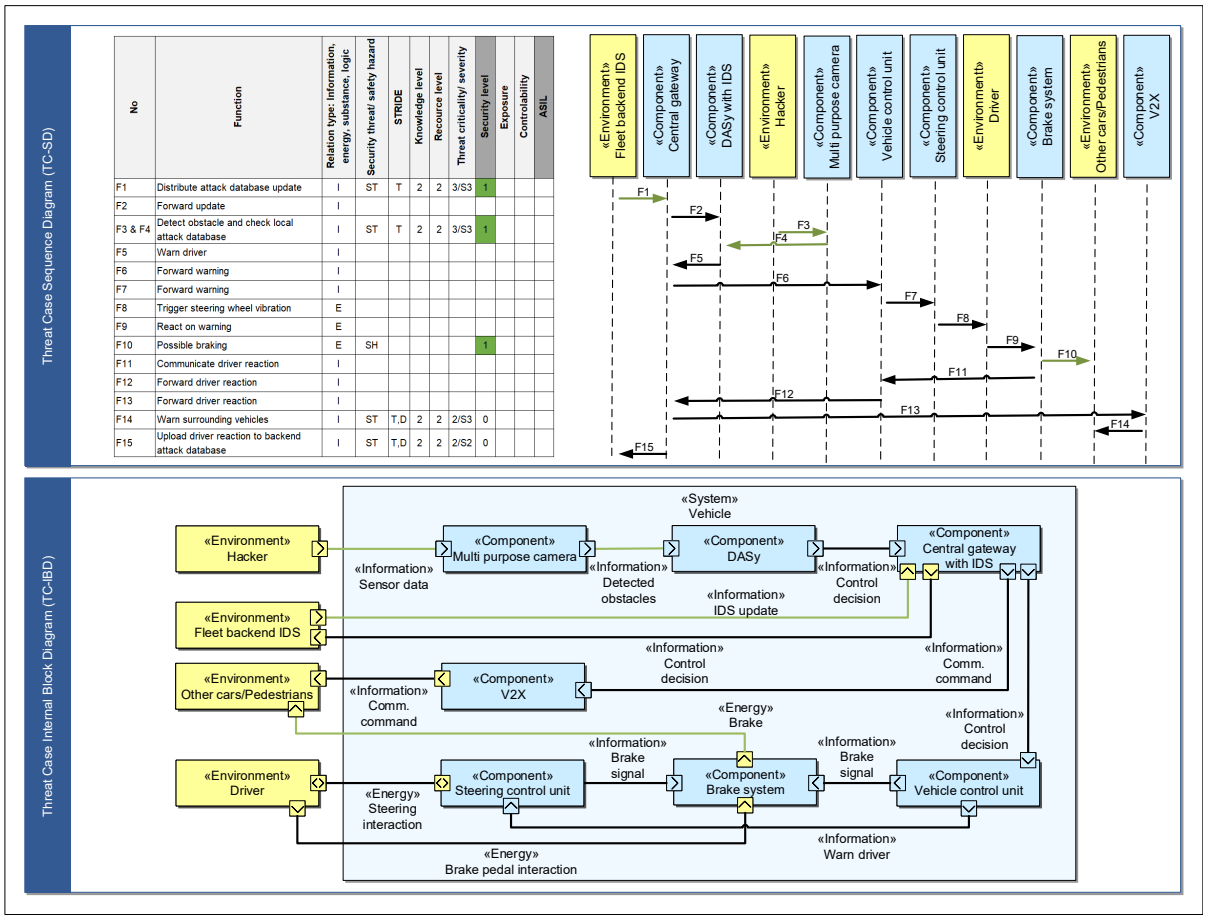
In this section, we present how identified security hazards can be resolved in the workshop using design patterns. As a basis, we use the system model, which already contains identified security hazards. We illustrate the resolution of security hazards, by using the DIDS security hazard design pattern. This results in a re-edit of the system model in terms of structure and behavior. To resolve the identified security hazards in the system model, different design patterns can be used.

In the trivial case the design pattern "Sensor Fusion" can be applied. For this purpose a further redundant sensor like a "Mid-range radar" is added to the system model. By using additional sensors the information quality is improved, so obstacles can be detected better. Due to the high degree of artificial intelligence and networking of autonomous vehicles and platoons and for better illustration of our method we will use the presented DIDS design pattern. The idea of the DIDS design pattern is to use an attack database that the system accesses. In this process, the data determined are compared with the data stored in the database. In case of anomalies the system is warned.

The DIDS design pattern uses a local database which synchronizes with a central database. To resolve security hazards in our system model, sensor data from the "Multi purpose camera" must be compared with data from the attack database (cf. Figure 3, TC-SD F3-F4). If in example the platoon leader (local attack database) or the vehicles of the fleet (central attack database) have detected previously placed cardboard bags as uncritical obstacles, a false alarm can be prevented (F8 & F 14). For the re-edit of the TC-IBD in Figure 3, we use the system structure IBD of the DIDS design pattern. We identify the component "Central gateway" as relevant for the DIDS extension. All data converge in this component. Therefore, the extension of this component to include an attack database is suitable. This allows incoming data to be directly compared with attacks from the database (F3-F4). For the synchronisation of identified attacks between different vehicles, we add the environmental object "Fleet backend IDS" to the system model. The "Central gateways" of the individual vehicles synchronize with the "Fleet backend" regarding identified attacks (F1 & F15). In order to help us to re-edit the TC-SD, we use the sequence diagram of the DIDS design pattern. Based on the extended TC-IBD, we extend the already created TC-SD with the new IDS components. In addition, we add new functions.

https://doi.org/10.1017/pds.2021.517 Published online by Cambridge University Press

After applying the DIDS pattern, the following function sequence results: The local attack database is synchronised at regular intervals (F1). If an obstacle is detected, the data is compared with the local attack database (F3 & F4). In the negative case, the driver is warned by vibration on the steering wheel that an obstacle has been detected on the road (F5 & F8). If the driver reacts, e.g. by braking (F9), the information is communicated to surrounding vehicles (platoon members) (F14). Based on the driver's reaction, an evaluation of the detected object as dangerous or as not dangerous and a synchronisation with the central attack database takes place. By applying the design pattern, new components, functions and relationships were added to the system model. A subsequent threat analysis using phase one of our method, shows that the application of the DIDS design pattern had an impact on the security and safety level for the set of functions. In this case the security and safety level of the functions has been reduced (cf. Functions F1, F3, F4 & F10). In addition, new security threats with F14 and F15 have appeared, which have been evaluated as non-critical. Overall, the security & safety rating serves as a decision support for the stakeholders. If the rating decreases after the application of a design pattern, the application was successful. If the rating increases, the stakeholders must decide whether the application of the design pattern must be discarded or whether another design pattern must be applied to solve the problem.



No	Function	Relation type: Information, energy, substance, logic	Security threat/ safety hazard	STRIDE	Knowledge level	Recource level	Threat criticality/ severity	Security level	Exposure	Controlability	ASIL
F1	Distribute attack database update	I	ST	T	2	2	3/S3	1			
F2	Forward update	I									
F3 & F4	Detect obstacle and check local attack database	I	ST	T	2	2	3/S3	1			
F5	Warn driver	I									
F6	Forward warning	I									
F7	Forward warning	I									
F8	Trigger steering wheel vibration	E									
F9	React on warning	E									
F10	Possible braking	E	SH					1			
F11	Communicate driver reaction	I									
F12	Forward driver reaction	I									
F13	Forward driver reaction	I									
F14	Warn surrounding vehicles	I	ST	T,D	2	2	2/S3	0			
F15	Upload driver reaction to backend attack database	I	ST	T,D	2	2	2/S2	0			

Figure 3. Resolution of security hazards in the initial system model by applying the DIDS design pattern

5 EVALUATION & DISCUSSION

Evaluation setting: In the context of a project at the University of Paderborn, we tested the described method with 67 interdisciplinary Master's students from the area of computer science, computer engineering and business informatics. For this, we developed the individual steps and tested them with a group of 3 people in advance. The project was carried out using platooning as an example and run over 8 weeks with 80 hours per student (total workload 5360h). Every week we held an hour-long video conference in which we asked about the status of the project and discussed uncertainties regarding the

https://doi.org/10.1017/pds.2021.517 Published online by Cambridge University Press

application of our method. In total 21 groups of about 3 persons per group used RE-EDIT to identify and resolve safety relevant security threats in the system design phase. On average, there were students from different courses in each team, so that for the most part there were interdisciplinary teams. The individual team workshops were conducted virtually in Microsoft Teams with Microsoft PowerPoint as the modelling tool. For this, we provided SysML templates for Powerpoint. The project had the following phases: (P1) Identification and visualisation of user stories and identification of security hazards in a black box model. To visualize threats from a black box perspective, the students used the MBSE tool 3D Engineer (Japs et al., 2020). (P2) Refinement of the black box model to a white box model and refinement of the identified security hazards. (P3) Selection of suitable design patterns and resolution of security hazards.

Evaluation goal: The aim of the evaluation by the project was to check the applicability of the method in workshops and to find potential for improvement for future work. In particular, we wanted to find out through the project which steps/tools/constructs need to be adapted in order to increase the effort/benefit ratio of applying our method (cf. Discussion). In addition, we wanted to find out what effects the application of our method had on the artefacts created (cf. Evaluation results).

Evaluation results: In Figure 4 we present the results of our evaluation of our method. Altogether we examined 658 slides from a quantitative point of view. With regard to qualitative aspects, our statements are based on the weekly video conferences. The complete data set of the 21 examined project results can be analyzed in detail under (Japs, 2020b).

	Group	Group ID	1	2	3	6	8	9	10	11	12	13	26	27	28	29	30	31	32	33	34	35	36	Average	Median
		Group size	3	3	3	3	3	4	3	3	4	3	2	3	3	4	3	3	3	4	4	3	3		
Black box view - Identify threats (40 h)	No of models	Use cases	4	6	6	4	4	6	4	4	6	4	2	4	5	6	8	4	8	7	7	5	6	5	5
		Threat cases	4	4	3	4	5	8	5	3	6	5	4	5	6	7	8	5	8	7	8	6	9	5,7	5
White box view - Identify & refine threats (20h)	No of models	Use cases	5	6	4	4	4	6	4	4	6	4	2	4	4	6	4	4	8	6	7	5	4	4,8	4
		Threat cases	8	6	5	6	5	8	5	4	7	5	3	5	8	7	8	5	7	7	8	6	6	6,1	6
	Archi-tecture	Relationships	15	46	18	27	27	27	11	15	24	23	21	17	8	22	39	21	16	10	11	41	20	21,9	21
		Components	14	31	15	14	18	22	11	16	20	17	18	5	8	17	26	19	15	11	12	22	19	16,7	17
White box view - Resolve threats (20h)	No of models	Use cases	6	6	4	4	4	6	4	4	6	4	2	4	5	6	4	4	8	5	7	4	4	4,8	4
		Threat cases	7	6	5	6	5	8	5	5	7	5	3	5	7	7	8	5	8	7	8	6	6	6,1	6
	Archi-tecture	Relationships	18	57	25	32	31	28	29	21	28	25	27	27	9	26	42	20	19	12	12	40	30	26,6	27
		Components	16	33	21	16	18	24	25	17	24	19	26	18	10	18	29	19	17	13	13	23	25	20,2	19
	Impact of REDIT	**New relation.**	3	11	7	5	4	1	18	6	4	2	6	10	1	4	3	-1	3	2	1	-1	10	**4,7**	**4**
		New comp.	2	2	6	2	0	2	14	1	4	2	8	13	2	1	3	0	2	2	1	1	6	**3,5**	**2**
		Changed relation.	7	0	4	0	5	2	0	3	0	0	0	4	1	4	3	0	0	3	0	2	0	**1,8**	**1**
		Changed comp.	0	11	4	0	0	4	5	0	0	1	0	0	0	3	0	0	0	0	0	0	0	**1,3**	**0**

Figure 4. Application of RE-EDIT by 67 interdisciplinary master students

In P1 in average 5,7 threat cases (TC-SDs) were created within 40 hours per group. Stereotypes were used to mark identified security and safety relevant functions and offered a basis for a common discussion within the groups. In P2 which lasted 20 hours, each group has derived an initial TC-IDB and created in average 6,1 threat cases (TC-SDs). The methods SAHARA and ASIL were used to identify safety critical functions. The TC-SDs were used to aggregate threats in the TC-IBD. On average, the TC-IBD of the 21 group projects analysed consists of 16.7 system components (internal system components & environmental objects) and 21.9 relationships between the components. In P3 which lasted 20 hours, each group applied design patterns to resolve the identified threats. On average, 6.1 threat cases (TC-SDs) were affected by the re-edit. This resulted in an expansion of the TC-IBD. RE-EDIT was used to expand or adapt an average of 4.8 (23.8 %) of the components and 6.5 (24.4 %) of the relationships in the TC-IBD to resolve the threats.

Discussion: Overall, using RE-EDIT, all teams were able to identify security hazards in joint workshops and solve most of them using the design patterns provided. All teams were able to successfully use the workshop applicable SysML constructs we developed for modelling in PowerPoint. On the one hand, PowerPoint was very easy to get started with, but it lacked functions that supported modelling, such as the automatic alignment of several model elements. The application of SAHARA and ASIL could be carried out without any problems using the tables provided, but the manual determination of the security and safety levels for the high number of functions was time-consuming. We did not explain exactly how to work together in the workshop, so some teams could not always work effectively. There was an understanding that in the workshop all participants must always work on one task. This meant

https://doi.org/10.1017/pds.2021.517 Published online by Cambridge University Press

that several groups missed the opportunity to work on several TC-SDs in parallel in subgroups. The use of explicit security and safety stereotypes to make a clear distinction was helpful, whereas the use of the CONSENS stereotypes Information, Energy etc. did not add any clear value. The choice of security hazard design patterns was severely limited for the participants, as we could only provide three workshop-appropriate design patterns at that time (Sensor Fusion, DIDS & Defense in Depth). The specified time frame of 80h (approx. 10 working days) for several workshop appointments, in particular for several stakeholders, is unsuitable for use in an industrial environment and too costly. A time frame of 5 working days would be more realistic. In addition to optimized templates, this is made possible in particular by focusing on especially safety-critical TC-SDs.

6 CONCLUSIONS & FUTURE WORK

In this paper, we presented our RE-EDIT method using autonomous driving as an example. RE-EDIT supports an interdisciplinary team of stakeholders in a workshop to identify security hazards and to resolve them with the help of design patterns. For easy threat modeling in the workshop, we created a lightweight security SysML profile. The method consists of the following steps: Modeling of threat cases in the form of SysML sequence diagrams; Risk assessment of threat cases; Derivation of a system architecture in the form of a SysML IBD and selection and application of security hazard design patterns for security critical threats. In a project led by us with 67 interdisciplinary master students, we were able to initially test the applicability and effectiveness of RE-EDIT on different use cases from the field of autonomous driving over a period of 8 weeks.
Part of future work is the application of RE-EDIT in workshops with industry participants. For this purpose, we additionally plan to develop a security hazard design pattern catalogue. Furthermore, we want to adapt the risk assessment to the soon to be published ISO 21434 "Road vehicles - Cybersecurity engineering".

7 ACKNOWLEDGEMENTS

This research was funded by the German Federal Ministry of Education and Research (BMBF) in the project SecForCARs, grant number 16KIS0790. The contents of this publication are the sole responsibility of the authors. We thank our students Oliver von Heißen, Sebastian Schmidt and Farnaz Ariafar for their support.

REFERENCES

Anacker, H., Dumitrescu, R., Kharatyan, A., Lipsmeier, A. (2020), "Pattern based systems engineering - Application of solution patterns in the design of intelligent technical systems", [16th International design conference, Cavat, Dubrovnik, Croatia], 10.1017/dsd.2020.107.

Amorim et al. (2017), "Systematic pattern approach for safety and security co-engineering in the automotive domain", [International Conference on Computer Safety, Reliability, and Security (SAFECOMP 2017), Trento, Italy], 10.1007/978-3-319-66266-4-22.

Bosch (2020), Products and Services, https://www.bosch-mobility-solutions.com/en/, last access: 2020-12-08.

Cheng, B.H.C., Doherty, B., Polanco, N., Pasco, M. (2019), "Security patterns for automotive systems", [ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems (MODELS-C), Munich, Germany], 10.1109/MODELS-C.2019.00014.

Dori, D. (2016), "Model-Based Systems Engineering with OPM and SysML", Springer.

Dumitrescu, R., Kaiser, L., Meyer, M. and Holtmann, J. (2013),"Automatic verification of modeling rules in systems engineering for mechatronic systems",[Proceedings of the ASME 2013 International Design Engineering Technical Conferences, Portland, Oregon, USA, 2013], 10.1115/DETC2013-12330.

Fernandez-Buglioni, E. (2013), "Security Patterns in Practice: Designing Secure Architectures Using Software Patterns", Wiley.

Gausemeier, J., Rammig, F.J., Schäfer, W. (2014), “Design methodology for intelligent technical systems”, Springer, Berlin-Heidelberg, 2014,10.1007/978-3-642-45435-6.

Greenberg, A. (2015), “Hackers remotely kill a jeep on the highway - with me in it”, https://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/, Wired Online, last access: 2020-09-11.

Goldman, D. (2015), “Chrysler recalls 1.4 million hackable cars”, 2015, CNN Business https://money.cnn.com/2015/07/24/technology/chrysler-hack-recall/index.html, last access: 2020-12-08.

https://doi.org/10.1017/pds.2021.517 Published online by Cambridge University Press

Heisel, M., Maidl, M., Wagner, M., Wirtz, R., Zhao, T. (2019), "Pattern-based modeling of cyber-physical systems for analyzing security", [24th European Conference on Pattern Languages of Programs (EuroPLoP'19), Irsee, Germany], 10.1145/3361149.3361172.
Howard, M., Lipner, S. (2006), "The security development lifecycle", Microsoft Press, 10.1109/MSP.2017.14.
International standards organization (2018), "ISO 26262: Road vehicles – Functional safety".
International standards organization (2015), "ISO/IEC/IEEE 15288:2015 Systems and software engineering — System life cycle processes".
Japs, S., Kharatyan, A., Kaiser, L., Dumitrescu, R. (2020), "Method for 3D-environment driven domain knowledge elicitation and system model generation", [16th International design conference, Cavat, Dubrovnik, Croatia], 10.1017/dsd.2020.41.
Japs, S. (2020), "Security & Safety by Model-based Requirements Engineering",[28th IEEE International Requirements Engineering Conference], 10.1109/RE48521.2020.00062,
Japs, S., Anacker, H., Dumitrescu, R. (2020), "SAVE: Security & safety by model-based systems engineering on theexample of automotive industry" (in press), https://owncloud.fraunhofer.de/index.php/s/hEuRbag1slmtyTD, last access: 2020-12-08
Macher, G., Sporer, H., Berlach, R., Armengaud, E., Kreiner, C.(2015), "SAHARA: A Security-Aware Hazard and Risk Analysis Method", [Proceedings of design, automation & test in europe conference & exhibition (DATE), Grenoble, France], 10.7873/DATE.2015.0622.
Mead, N.R., Stehney, T. (2005), "Security quality requirements engineering (SQUARE) methodology", [Proceedings of the 2005 Workshop on Software Engineering for Secure Systems—Building Trustworthy Applications,St. Louis, Missouri], 10.1145/1082983.1083214.
Pohl, K. (2016), "Requirements engineering: Fundamentals, principles, and techniques", Springer.
Rehman, S. U., Allgaier, C., Gruhn, V. (2018), "Security requirements engineering: A framework for cyber-physical systems", [2018 International Conference on Frontiers of Information Technology (FIT), Islamabad, Pakistan], 10.17185/duepublico/71232.
Rupp, C. and die SOPHISTen (2014), "Requirements-Engineering und -Management: Aus der Praxis von klassisch bis agil", Carl Hanser Verlag.
SAE International (2016), "Cybersecurity guidebook for cyber-physical vehicle systems J3061".
SecForCARs 2020, Research project security for connected automated cars, https://www.secforcars.de, last access: 2020-12-08.
Shostack, A. (2014),"Threat Modeling: Designing for Security",Wiley,2014.
Stølen, K. et al. (2002), "Model-based risk assessment - the CORAS approach", [Proceddings of the norsk informatikkkonferanse (NIK 2002), Tapir].
CONSENS application (2020),https://smartmechatronics.de/consens, last access: 2020-12-08.
Data set (2020), https://owncloud.fraunhofer.de/ index.php/s/01hyejztC0nrRJ8, last access: 2020-12-08
Object Management Group (2015), Systems modeling language specification V. 1.4s.
Unity Consens (2020), Project References on the UNITY Innovation Alliance, https://www.unity-innovation-alliance.com/en/, last access: 2020-12-08.
Tekaat, J., Kharatyan, A., Anacker, H., Dumitrescu, R.(2019),"Potentials for the integration of design thinking along automotive systems engineering focusing security and safety",[International Conference on Engineering Design (ICED), Delft, The Netherlands], 10.1017/dsi.2019.295.

https://doi.org/10.1017/pds.2021.517 Published online by Cambridge University Press