A Method for Identifying Inconsistencies between Functional Safety and Cybersecurity of Autonomous Vehicles

Priyadarshini -, Simon Greiner, Maike Massierer, Oum-El-Kheir Aktouf

A Method for Identifying Inconsistencies between Functional Safety and Cybersecurity of Autonomous Vehicles

Priyadarshini*†, Simon Greiner*, Maike Massierer*, Oum-El-Kheir Aktouf†
*Robert Bosch GmbH, Cross-Domain Computing Solutions, Abstatt, Germany
Email:{priyadarshini.priyadarshini,simon.greiner,maike.massierer}@de.bosch.com
†Univ. Grenoble Alpes, Grenoble INP, LCIS, Valence, France
Email: {priyadarshini.-,oum-el-kheir.aktouf}@lcis.grenoble-inp.fr

***Abstract*—With the shift towards assisted and autonomous driving, more functional safety and security features are being integrated to make these systems dependable, safe, and secure. Additional safety and security features lead to increasing interdependencies and inconsistencies between them. In this paper, we propose a method to detect inconsistencies between safety and security by assessing the potential impact of security features on safety-critical components of a system. For the assessment, we analyze artifacts from the Failure Mode and Effects Analysis and SysML models of the overall system. We apply our method to an example, i.e. we evaluate whether the security feature *secure onboard communication* (AUTOSAR SecOC) could potentially cause a failure of the safety-critical communication. Our method identifies potential interactions between safety-critical components and security features by reusing available functional safety and security artifacts generated as a part of their independent analyses.**

***Index Terms*—functional safety, cybersecurity, autonomous driving, autonomous vehicles, interdependencies, interferences, inconsistencies, system FMEA, SysML activity diagram, SysML internal block diagram**

I. Introduction

Automotive systems are currently changing at a fast pace. On the one hand, the complexity of safety-relevant vehicle functionalities is growing quickly due to rapid progress in automation of driving functions. On the other hand, an increase in connectivity of vehicles leads to larger attack surfaces, which in turn requires more and more protection mechanisms against cyberattacks.

Safety engineering is an established discipline in automotive engineering and regulated mainly in the automotive functional safety standard ISO 26262 [1]. In comparison, regulation for cybersecurity is relatively new, and the upcoming road vehicle cybersecurity standard ISO/SAE FDIS 21434 [2] will most likely be the leading regulation. Both norms pose rigorous engineering requirements on processes and measures to protect the safety and security of vehicles. However, their interaction is limited, and a closely coupled co-engineering of safety and security engineering is not trivial, if sensible at all. As a result, safety and security engineering are practically performed in parallel without much interaction by experts in the respective fields during different phases of the development.

The distribution of safety and security engineering activities between different people and their division into separate tasks bears the risk that safety and security will interfere with each other. For example, a security analysis may require protection against attacks on communication (for instance in the form of firewalls or message authentication that might discard messages for security reasons), while safety relies on the assumption that every message received by a system is processed. The requirements resulting from both disciplines therefore can have unintended interdependencies. If these unintended interdependencies are detected too late during system development, e.g. during final testing, this can require changes of the overall system architecture. This is very expensive and can lead to delays in the start of production of a vehicle.

In this paper, we propose a method to identify conflicts between safety and security solutions resulting from the potential impact of security features on the safety-critical components of a system. We focus only on safety-critical communication interfaces, such as the Ethernet communication interface, as they are typical entry points for an attacker. This method is based on the artifacts created during safety and security engineering and on semi-formal models of the system architecture. Furthermore, the method provides the potential impact of identified inconsistencies on the system level. As a prototypical analysis, we have manually applied this method to a real subsystem that is illustrated with an example of a security feature called *secure onboard communication* (SecOC) [3], an automotive standard protocol for authentication of communication (Com) in vehicle bus systems.

This paper is structured as follows. In the following section, we shortly introduce safety and security related artifacts of respective analyses on which our method builds. In Section III, we introduce our method. In Section IV, we discuss related work, and finally, we conclude.

II. Functional safety and cybersecurity analysis

We assume that functional safety and security analyses have been performed on the system of interest. We reuse the artifacts of these analyses in our method. In particular, we briefly introduce the system level failure modes and effects

analysis (FMEA) and system modeling language (SysML) models of security features on which our method is based.

A. System FMEA

A system FMEA [4], [5, Chapter 4] is a structured approach for systematic analysis of potential failures of a system, so-called failure modes. It has five steps: structure analysis, function analysis, failure analysis, actions analysis, and optimization. We limit our description to the first three steps, since only these are required for our method.

1) Structure analysis: A structure analysis diagram shows a hierarchical order of a system and its *system elements*, including their interfaces.

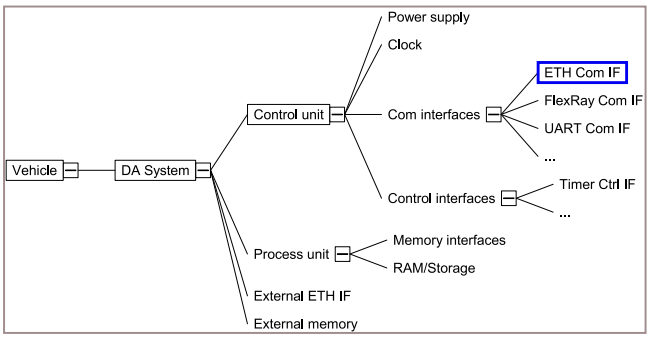

Fig. 1. FMEA structure analysis - an example of a system structure

For example, Fig. 1 shows a simplified example of a structure analysis diagram for a *Vehicle*. It is composed of a driver assistance system called *DA System*, which comprises different system elements, such as a *Control unit* and a *Process unit*. These system elements can have different interfaces, such as the *Com interfaces*. These can be further classified as the Ethernet (ETH), FlexRay, and UART communication interfaces (IF), shown as *ETH Com IF*, *FlexRay Com IF*, and *UART Com IF*, respectively.

2) Function analysis: Function analysis provides an overview of the functions and sub-functions implemented by each system element. Additionally, it highlights the cause-and-effect relationships between functions and sub-functions.

3) Failure analysis: In this step, potential failure modes are deduced from each function of every system element, stating what can go wrong for each function and sub-function.

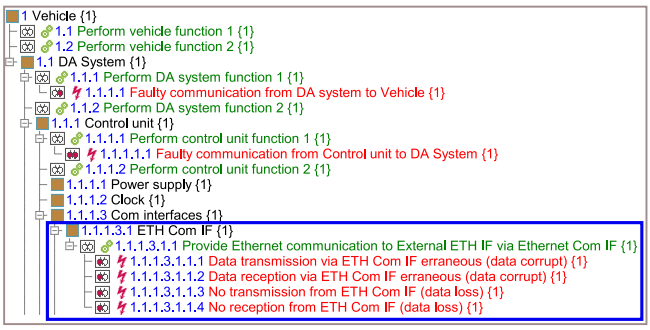

Fig. 2. FMEA function and failure analysis

In Fig. 2, the failure analysis for the *ETH Com IF* shows that this system element implements the function of providing external communication (green text). For this function, four different failure modes have been identified (red text).

Failure structures show how failure modes of different system elements are related to each other. For example, the failure net depicted in Fig. 3 shows how a failure mode of the *ETH Com IF* leads to a failure mode in the *Control unit*, which itself causes a failure mode in the *DA System*. In the notion of failure analysis, the failure structure shows *causes* and *effects* for each failure mode.

B. SysML models of a security feature

We assume that SysML [6] models exist for the security features implemented by the system. In particular, we assume a structural model of each security feature and its environment in the form of an internal block diagram and a behavioral model of the security feature as an activity diagram.

1) Internal block diagram (IBD): A block in SysML represents a system, a hardware, or a software component. An internal block diagram is a SysML structure diagram that shows the encapsulated structural contents (parts, properties, connectors, ports, interfaces) for a given block.

For example, Fig. 4 shows that the *Control unit* consists of the *ETH Hardware Software Interface (HSI)*, the *Secure Onboard Communication*, and the *Hardware Security Module* components. Further, it shows that the *Control unit* is connected to the external Ethernet interface communication module called *External ETH IF* via the *ETH Com IF* ports.

2) Activity diagram: A SysML activity diagram defines the behavior of the block it belongs to. It defines the functional behavior and control flow in the form of actions and conditions for a block. For example, Fig. 5 shows an activity diagram for the block secure onboard communication.

Initially, the secure onboard communication block performs four *actions* (orange rounded rectangles), and then – depending on the evaluation of a *condition* (green diamond) – it performs another action before the activity terminates.

III. A Method for Impact Analysis of Security on Safety

In this section, we propose a method to assess the potential impact of the functional behavior of a security feature on the safety-critical components of a system. Our method consists of three steps. In the first step, for each interface mentioned in the FMEA structure analysis we identify related security features. In the second step, we analyze whether the functional behavior of the security feature can lead to a failure mode related to the interface. In the third step, we identify the impact of the failure mode caused by the security feature.

To illustrate our method, we use the security feature secure onboard communication (SecOC) and a safety-critical Ethernet (ETH) communication interface as a running example. SecOC is defined by the AUTomotive Open System ARchitecture (AUTOSAR) [7] and verifies the freshness and authenticity of messages communicated over an Ethernet bus.

Fig. 3. Example of an FMEA failure net

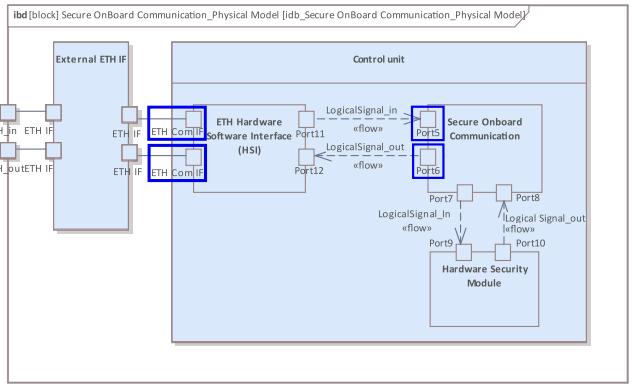

Fig. 4. Example of an internal block diagram

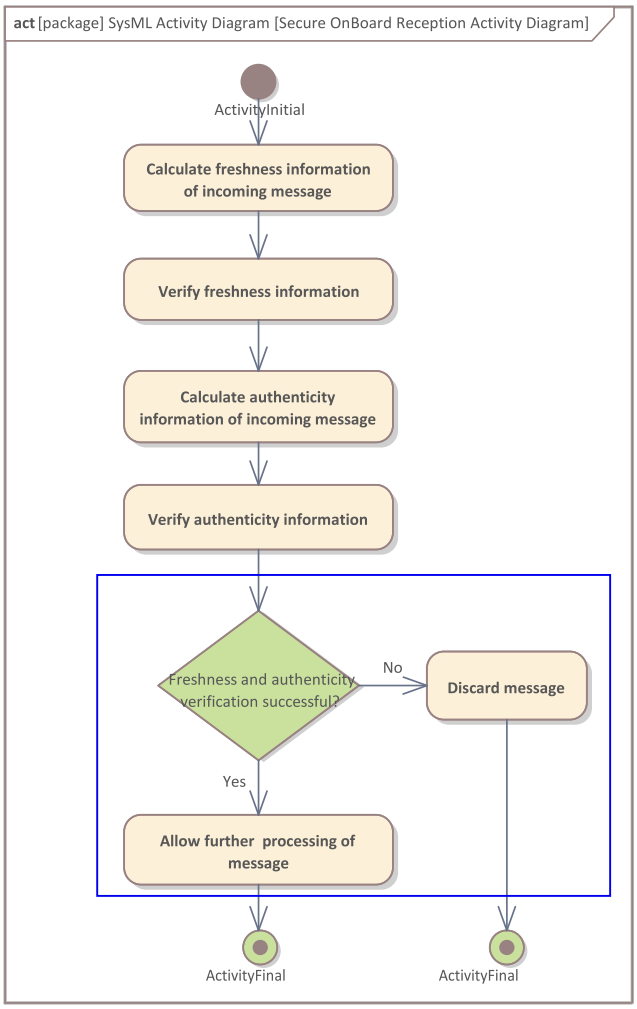

Fig. 5. Example of an activity diagram

A. Step 1: relation of security features with interfaces in the system FMEA

Communication interfaces provided by a system to its environment are typically an entry point for attackers. Therefore, many security features protecting against such attacks have an influence on the communication via these interfaces. At the same time, many safety-relevant malfunctions of automotive embedded systems have an effect on the communication via an interface (e.g. data corruption, omission, or wrong timing of messages). For this reason, we use communication interfaces as the initial connection between the safety analysis and the model of the security features.

In the first step of our method, we collect all interfaces with failure modes assigned to them in the FMEA structure analysis. For each of these interfaces we identify the corresponding port in the SysML model of the system. For the port, we collect all blocks modeling security features that have a data flow to or from this port. As a result, we know the port over which the security feature could intervene with the safety-relevant communication interface.

In our example, we first choose the *ETH Com IF* highlighted in Fig. 1. The corresponding port in the SysML model is the port *ETH Com IF* of the *Control unit* highlighted in Fig. 4. By analyzing the data flow in the SysML model, we can identify that the *Secure Onboard Communication* block can intervene with communication on the *ETH Com IF* port via *Port 5* and *Port 6*.

B. Step 2: detection of potential inconsistencies

In the second step, we analyze whether the control flow of the security feature has an influence on the communication over the port(s) identified in the previous step. We also identify the failure modes of the interface selected in step 1 that this influence corresponds to. We use activity diagrams to perform the control flow analysis and the FMEA failure analysis to identify the corresponding failure mode(s). The potential inconsistency is identified by analyzing whether the control flow of the security feature could cause one or more failure modes on the identified port in step 1. We collect these failure modes to assess their impact in step 3.

In our example, the activity diagram of the secure onboard communication block shows that the received messages are discarded depending on the result of a freshness and authenticity check (green diamond in Fig. 5). Therefore, there is the potential that a safety-critical message is lost, which is equivalent to the failure mode *no reception from ETH com IF (data loss)* in Fig. 2.

C. Step 3: impact analysis of detected inconsistencies

In the third step, we identify the potential impact of the interference detected in the previous step. There, we identified the failure mode that could be caused by the security feature. We now use the failure net of the identified failure mode to analyze its failure effects on different elements of the system.

In our example, the failure net in Fig. 3 shows that the identified failure mode *no reception* can lead to the effects *faulty communication* on the level of the *Control unit* and *faulty communication* on the level of the *DA System*. After identifying the potential conflict and its resulting impact, relevant stakeholders should be invited for reviewing and resolving the conflict.

IV. Related Work

In this section, we briefly review existing methods focussing on the identification, characterization, and resolution of safety and security interdependencies. Four types of interdependencies between safety and security are identified in [8], namely *conditional*, *reinforcement*, *antagonism*, and *independence*. Safety and security could each act as a condition for the other, complement each other, conflict with each other when considered separately for the same system, or be independent of each other. In this paper, we focus on conflicts resulting from inconsistent safety and security features specified in the corresponding safety and security requirements that are allocated to the same system.

Different approaches have been explored in the literature to master these interdependencies, such as formal methods and model-based methods. Piètre-Cambacédès and Bouissou [8] propose an approach based on a Boolean logic driven Markov processes (BDMP) dynamic formalism to graphically model risk scenarios and characterize safety and security interdependencies. Martin et al. [9] focus on a consideration of security aspects within a safety engineering lifecycle and propose a systematic pattern-based approach that interlinks safety and security patterns. Gu, Lu, and Li [10] analyze and classify the relationship between safety and security requirements in industrial control systems as *interdependent*, *conflicted*, and *unrelated*. In [11] and [10], the authors propose a formal framework to analyze conflicts in security and safety requirements.

Although some of these works focus on the identification of safety and security conflicts in the requirements phase, we argue that the requirements themselves may not be conflicting, but still the resulting safety and security measures could be. Cui et al. [12] point out the need for future research into the consistency between safety and security countermeasures when they are developed independent of each other. To harmonize the potential conflicts, bidirectional analysis is essential to assess the impact of security features on system safety and of safety mechanisms on system security. Our contribution is the proposed method to assess the potential impact of security features on critical safety components of a system by reusing the existing artifacts of functional safety and cybersecurity engineering.

V. Conclusion and Future Work

In this paper, we have presented a method to assess the potential impact of security features on the communication interfaces that are safety-critical and may serve as an entry point for attackers at the same time. This method does not combine functional safety and cybersecurity analysis as a joint method but rather checks for potential interactions between them by reusing available functional safety and cybersecurity artifacts generated as a part of their independent analyses. Since this method is performed at the system level, safety and security inconsistencies can be identified in early development phases to optimize resources and to resolve these inconsistencies through either trade-off or co-design and co-development.

In future work, we will develop a model-based automation tool to automate our proposed method. The tool will allow us to assess the impact of security features not only on defined safety-critical communication interfaces, but also on other system elements in the FMEA system structure, such as external memory. We will further investigate additional information and parameters that can support our analysis, whereby the identified security features with a potential impact on safety-critical system elements will also be traceable to the corresponding system security requirements.

Acknowledgment

This work is supported by the French National Research Agency in the framework of the "Investissements d'avenir" program (ANR-15-IDEX-02).

References

[1] International Organization for Standardization, "ISO 26262: Road vehicles – Functional Safety," 2011.

[2] ——, "ISO/SAE FDIS 21434: Road vehicles – Cybersecurity engineering." [Online]. Available: https://www.iso.org/standard/70918.html

[3] AUTOSAR, "Specification of Secure Onboard Communication, Release 4.3.1," Dec 2017.

[4] VDA-QMC, "VDA Volume 4 – Quality Assurance before series production Part 2," Nov 1999.

[5] VDA-QMC, "VDA Volume 4 – Quality Assurance in the Process Landscape," June 2012.

[6] "SysML Open Source Project – What is SysML? Who created it?" [Online]. Available: https://sysml.org/

[7] "AUTOSAR." [Online]. Available: https://www.autosar.org/

[8] L. Piètre-Cambacédès and M. Bouissou, "Modeling safety and security interdependencies with BDMP (Boolean Logic Driven Markov Processes)," in *2010 IEEE International Conference on Systems, Man and Cybernetics*. IEEE, 2010, pp. 2852–2861.

[9] H. Martin, Z. Ma, C. Schmittner, B. Winkler, M. Krammer, D. Schneider, T. Amorim, G. Macher, and C. Kreiner, "Combined automotive safety and security pattern engineering approach," *Reliability Engineering & System Safety*, vol. 198, 2020.

[10] T. Gu, M. Lu, and L. Li, "Extracting Interdependent Requirements and Resolving Conflicted Requirements of Safety and Security for Industrial Control Systems," in *2015 First International Conference on Reliability Systems Engineering (ICRSE)*. IEEE, 2015, pp. 1–8.

[11] M. Sun, S. Mohan, L. Sha, and C. Gunter, "Addressing Safety and Security Contradictions in Cyber-Physical Systems," in *Proceedings of the 1st Workshop on Future Directions in Cyber-Physical Systems Security (CPSSW09)*, 2009.

[12] J. Cui, L. S. Liew, G. Sabaliauskaite, and F. Zhou, "A review on safety failures, security attacks, and available countermeasures for autonomous vehicles," *Ad Hoc Networks*, vol. 90, 2019.