The Security for Safety Problem in Cyberphysical Systems

Semen Kort, Ekaterina Rudina
Kaspersky Lab, November 2015

Abstract

This paper is intended to clarify the key differences and identify similarities between security and safety in cyberphysical systems in order to propose an approach that addresses both types of issues present in these systems. Special attention is given to the *Security for Safety problem*, and an approach to threat modeling for this concept is provided. Finally, a variant of MILS-based architectural support is proposed for mechanisms implementing Security for Safety.

Keywords

Cyberphysical systems, security, functional safety, threat modeling, MILS, design approach

Motivation and research objectives

The motivation for this research arose from continued disputes about the validity of using cybersecurity methods to enhance the safety of cyberphysical systems. The old belief that safety mechanisms do not need hardening, even in cases when the system may interact with external untrusted systems and networks, remains strong. At the same time, the cybersecurity issues that arise for such systems are capable of affecting their functioning. How this correlates with issues that may cause immediate physical harm is not always clear.

There are several works devoted to this correlation. Pietre-Cambacedes and Claude [1] analyze the relations between safety and security properties, their differences and similarities. Sabaliauskaite and Mathur [2] propose an approach for safety and security integration based on the corresponding security and safety lifecycles. In Fovino et al. [3] a method for a quantitative security risk assessment is presented that combines fault-tree analysis traditionally used in reliability analysis, and attack-tree analysis proposed for the study of malicious attack patterns. Eames and Moffett [4] describe techniques for harmonizing safety and security requirements based on security and safety modeling, different documentation structures, the interaction of safety and security requirements, and the isolation of safety and security requirements processes.

Existing works implicitly separate the notions of safety and security, when these notions should be considered jointly in the context of a common issue. This issue is explored ad hoc for known incidents where safety was violated as a result of (or along with) a security violation. At the same time, there are many examples where the security and safety aspects are not connected and should be addressed as usual with the appropriate methods. It is necessary to define the aforementioned issue and describe the relevant methods for system assessment and protection. Use of the MILS concept is reasonable because it is valid for both notions considered separately and together (as will be substantiated further).

The objectives of this research are to:

- Describe the types of possible information security issues and functional safety hazards and their relationship in cyberphysical systems.
- Determine what types of issues in cyberphysical system design and implementation may cause safety problems in the event of cyberattacks, and under what conditions.
- Provide a unified approach to security threat modeling that takes into consideration both the informational threats and physical hazards that may be caused by possible attacks.
- Propose an approach for the architectural design of a system resistant to the safety issues that may arise from cyberattacks.

Classification of possible safety and security issues in cyberphysical systems

Traditionally, cyberattacks are considered to be issues that arise in the informational environment and target informational aspects of system execution. This leads to the interpretation of information security as the CIA (confidentiality, integrity, availability) set of aspects. Improper system behavior (e.g. software bugs, backdoors, Trojan programs) is also considered a source of problems that may affect these aspects and therefore relates to the information security scope.

Attempts to classify security threats to cyberphysical systems in the same way they are classified for pure IT systems lead to difficulties in describing the potential physical impact caused by a cyberattack. One example of this approach is to rearrange the CIA triad to AIC [5] by first ensuring the availability aspect in the control systems and attaching less importance to confidentiality. The availability aspect is important, but it alone cannot define all the physical characteristics that matter.

Cyberphysical systems exist in at least two types of environment: the informational environment and the physical environment. Therefore, issues may arise from both types of environment and affect physical aspects, informational aspects and the system itself (see Figure 1).

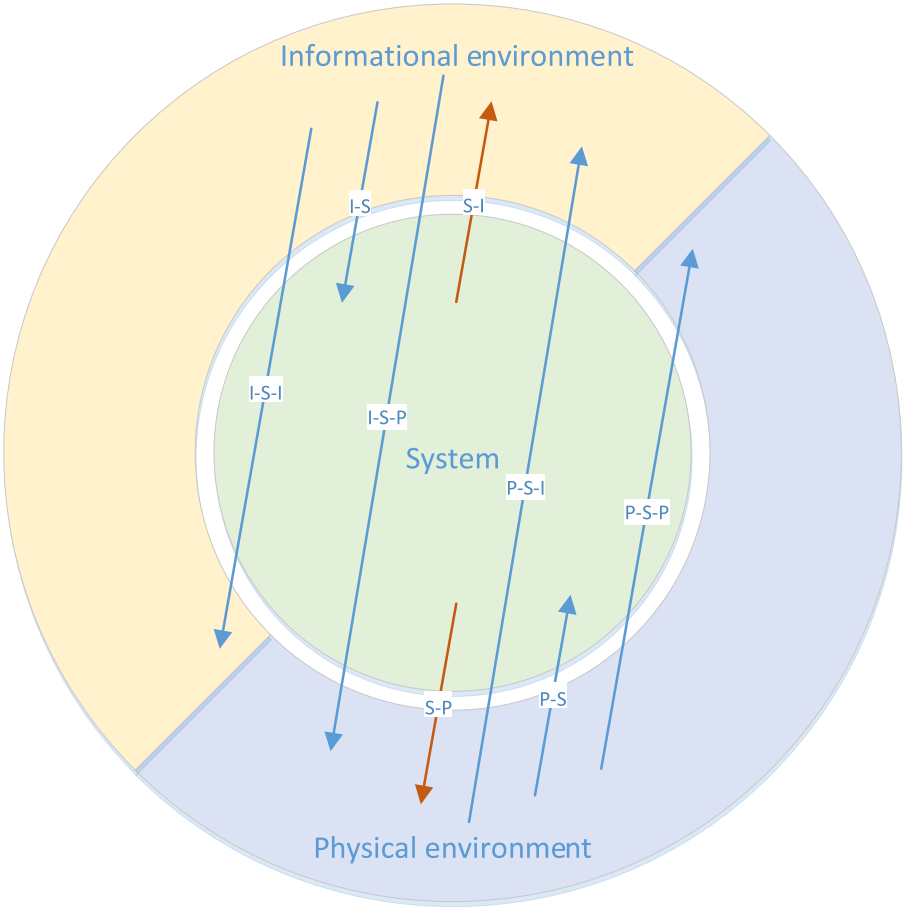

Figure 1- Possible impact vectors for cyberphysical systems

The potential impact and prevention measures may differ significantly for these issues. Firstly, let's provide examples for every type of impact vector (shown as arrows in Figure 1) to make them clearer.

The vectors that arise from the informational environment are cyberattacks. A cyberattack may target the system on its own (impact vector I-S), for example, to cause a denial of service or to steal confidential information. The attack may also target the informational environment of the system by exploiting improperly implemented system features (impact vector I-S-I). This vector is best illustrated by cross-site scripting (XSS) attacks. These two vectors are out of the scope of this research.

The vector I-S-P relates to cyberattacks targeting the physical environment of the cyberphysical system. Examples of such attacks are Stuxnet [6], an attack on an unnamed German steel mill facility [7], and the recent research by Miller and Valasek on car security [8]. The rest of this paper is mostly devoted to this impact vector.

The vector S-I includes software bugs or concealed system features capable of violating information security without external interference. This impact vector is usually considered to be an information security vector, though it would be more accurate to interpret it as information

safety vector (e.g., using a system infected with malware is unsafe). This impact vector is also outside the scope of this paper.

The vector S-P can, in a similar way, be associated with the functional safety of a system. If a system is not examined properly, its unsafe behavior may affect important factors in the physical environment. This impact vector and the applicable countermeasures are detailed below.

The vectors that arise in the physical environment are actually among those that are usually capable of harming the system or its components. The impact vectors P-S-P and P-S are usually mitigated by a set of physical, organizational, and deterrent measures. The next section takes a closer look at these measures. Of particular interest here is the vector P-S-I that refers to effects on information security by purely physical means. Besides the trivial examples of denial of service attacks caused by destroying hardware or cable breakage, this includes physical tampering of video surveillance systems by placing a picture in front of a camera. Although such attacks may be important, they are specific to the domain or environment of the system.

Based on these possible types of issues in cyberphysical systems, it is necessary to pay special attention to the I-S-P impact vector, reveal the conditions under which the existing methods of ensuring proper system behavior may be ineffective for this impact, and propose an appropriate approach to threat modeling that eliminates the relevant safety risks.

While the methods that guarantee safe system behavior are well known and have been applied for decades, these methods cannot actually give the same guarantees in the event of deliberate attempts to cause improper system behavior by external means. This is partially confirmed by the aforementioned incidents [6,7,8] when the safety controls do not prevent serious physical damage caused by cyberattacks.

Defining the Security for Safety problem

The most controversial issue from those listed above is the use of the I-S-P impact vector to affect the physical environment (in particular, to cause a safety violation) by exploiting system features or vulnerabilities. Let’s look at the scenarios where this impact cannot be mitigated by the same means that prevent the use of the P-S-P and S-P vectors.

Physical damage caused by physical means (P-S-P impact vector) is usually interpreted as human error, intentional system misuse or sabotage. To prevent these types of violation one can make organizational changes, apply deterrent measures and implement physical protection of system components and channels. These activities mostly restrict factors that are valid for a physical environment and cannot effectively reduce exposure to informational risks. However, the safety measures discussed below are partly applicable for mitigating the P-S-P vector.

Due to its own complexity, the system itself may affect the physical environment in an unacceptable manner. To minimize the risk of affecting the system via the S-P vector, functional safety guarantees for the system shall be obtained.

The functional safety notion is defined in the IEC 61508 standard [9] as follows: “Harm: physical injury or damage to the health of people either directly, or indirectly as a result of damage to property or to the environment. Risk: combination of the probability of occurrence of harm and the severity of that harm. Safety: freedom from unacceptable risk.” Functional safety is part of overall safety that depends on a system or equipment operating correctly in response to its inputs. Functional safety is the detection of a potentially dangerous condition resulting in the activation of

a protective or corrective device or mechanism to prevent hazardous events arising or providing mitigation to reduce the consequences of a hazardous event.

The methods and means of guaranteeing functional safety are important for our analysis. In some cases the I-S-P impact vector can be effectively eliminated by functional safety measures that were initially designed for the S-P vector; in some cases it can't. A shallow analysis of the latter reveals the following reasons for this ineffectiveness:

1. The safety measures are designed without taking into account the possibility of intentional violations. All safety violations are considered to be accidental events or a string of coincidences.
2. The set of system vulnerabilities that can be exploited by an external attacker (with the appropriate system exposure) is wider than the set of system defects that can of themselves and without any malicious factors lead to a safety violation.

We define the problem of protecting against dangerous impacts on system safety caused by cyberattacks as the Security for Safety (SfS) problem.

The sources of Security for Safety problems

The sources of Security for Safety problems lie in the informational environment of the system. It may be an external attacker, an unintentional mistake by a user, malicious use of the system by an insider, connected systems that are infected with malware, and so on. They are indistinguishable from the sources of cyberattacks in IT systems.

On the understanding that a cyberattack constitutes a special input (to exploit a vulnerability) intended to place the system into an unusual (insecure) state under particular conditions, two generic methods for keeping the system in a secure state may be described. The first is input validation, and the second is the monitoring of the system or its environment. As a result of this monitoring, the system, its components or data may be forced to return to a state that meets the necessary requirements. These methods are applicable universally to all types of systems and software. For the I-S-I impact vector example mentioned above – XSS attack – these methods are instantiated by the user input validation and application output encoding. Some protection solutions may implement these methods jointly. For example, anti-malware solutions may use signature detection as a method of input validation and behavioral analysis, which is a kind of monitoring of the system state. Generally, any technical protection method may be interpreted as a kind of input validation or output monitoring.

As for cyberphysical systems, these methods of maintaining proper system execution can be characterized as follows (see Figure 2).

1. The input validation mechanisms that work with data supplied by the informational environment can only currently prevent common security threats (such as malware infection or known vulnerability exploitation if the system is based on COTS software). This is not enough. Firstly, to affect the physical aspects of system functioning, one may apply special methods and exploit system features in an inappropriate way that is either not covered by an input surveillance mechanism or cannot be detected with validation methods due to a lack of knowledge about the physical nature of the process under attack. Secondly, taking measures according to the results of input analysis means interfering in running processes in an unexpected manner that may be unacceptable for some

cyberphysical systems. Thirdly, for special-purpose systems protection software may not exist.

2. Monitoring of the cyberphysical system is performed in the target – physical – environment. Implementation in the very diverse set of systems under consideration may vary, from the absence of such mechanisms to the highly reliable safety instrumented systems (SIS). Such systems were implemented primarily to guarantee the functional safety of process execution. Safety instrumented systems must be deployed independently from all other control systems that control the same equipment in order to ensure SIS functionality is not compromised. Most safety engineers would prefer there to be no integration between safety and control systems at all (see the 'Environment monitoring' and 'Safety enforcement' rectangles in Figure 2). Safety enforcement receives data from independently implemented or system-based monitoring mechanisms and performs the necessary actions in order to keep the system functioning within the necessary constraints.
   At the same time, experts state [10] that even for highly dangerous areas not all facilities adhere to the strict separation of safety and control for safety protection. Particularly, they mention: "Which architecture is best for a particular company depends on the organization's business strategy and tolerance for risk. At companies where safety at any cost is top priority, separate control and safety systems are likely to continue to remain the preferred approach. Companies looking to maximize cost savings are more likely to adopt a common platform integrated control and safety system".
   In any case, here we are considering not only the implementation of safety for critical systems in industrial automation, specified in full detail by the appropriate standards [9, 11], but also systems for which the common practices of monitoring and safety enforcement, with a necessary accent on the security aspect, still do not exist or are not widely applied (e.g., automotive systems, Internet of Things appliances, etc.).

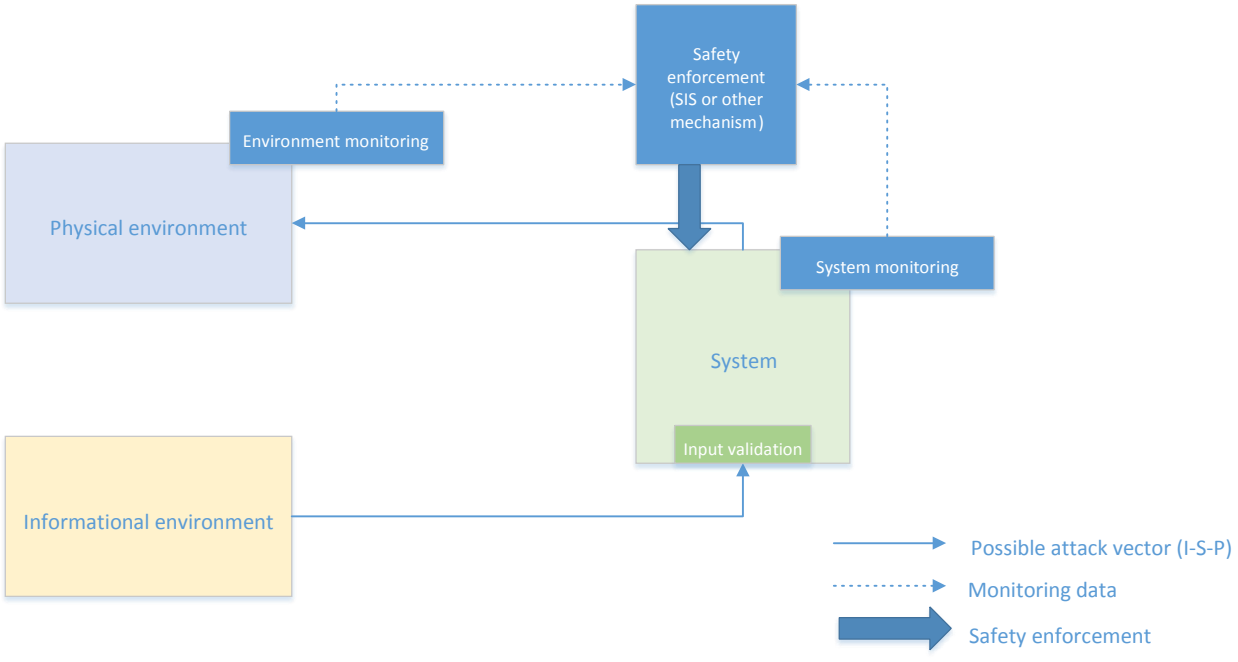

Figure 2 – The I-S-P impact vector with possible protection mechanisms

Where are the weak links in this chain? Let's study it step-by-step.

1) The aforementioned lack of, or inappropriate, input validation that might be used to monitor attempts of system misuse, attacks on the informational channel or on the user (social engineering, request forgery, etc.). If input validation is still implemented, it may be the target of an attack.
2) System monitoring may be disabled because of a successful attack. This is an argument for the use of detached (environment) monitoring, although in the case of an advanced persistent threat (APT) the external SIS may be deactivated. This is why experts emphasize the need for securing the Safety Instrumented System (SIS) [10].
3) The monitoring data may be tampered with to force the wrong decision about the current safety status. The complexity of this attack depends on the architecture of the system and safety mechanisms, but it should also be taken into account.
4) The safety enforcement mechanism may be disabled.
5) The channels used for safety enforcement may be compromised.

Using these conditions, let's define the approach to threat modeling for the Security for Safety problem.

The approach to Security for Safety threat modeling

An effective system-modeling technique should take into account both possible system flaws and threats of interest to the attacker. Threat modeling can be used to find out which vulnerabilities in system components are the most dangerous, and how they can be exploited by the attacker to violate the basic security aspects of the system and cause harm (in particular, to safety).

The validity of threat modeling results depends on how correct the assumptions were that were set before the modeling. Actually, this is true in both cases (for security and for safety). If safety mechanisms rely on assumedly reasonable actor behavior (even taking into account unintentional mistakes), a safety violation may be caused by an intentionally malicious actor. The phrase "rely on the reasonable behavior" means that the system assertions about the order of running operations and the parameters of these operations are not reinforced by system implementation. Security incidents also often happen according to an unforeseen system usage scenario. The attacker may break the assumptions made during threat modeling and security mechanism implementation in order to bypass this mechanism.

Threat modeling for the Security for Safety problem requires

- the revision of assumptions often made by safety engineers from a security point of view;
- consideration of possible weaknesses in the protection components listed in the previous section;
- a definition of an approach allowing the unification of security threats and safety hazards within a structured (possibly formal) description.

Fovino et al. [3] propose a unified approach based on merging the fault trees and attack trees. The appropriate method can be described as follows: attacks relevant to the safety failures are attached to these failures. Attack trees are incorporated into fault trees by the use of an OR gate, which indicates that a failure may be caused by intentional attacks. From the formal point of view, this approach works due to the uniformity of both representations.

It is also necessary to determine what types of attacks may lead to a safety violation, and how. Based on the analysis of possible gaps in Security for Safety enforcement, we may refine the method of threat modeling.

To do this, we need to use the relevant classification of possible threats. The widespread STRIDE model [12] is described below:

- S – Spoofing
- T – Tampering
- R – Repudiation of origin
- I – Information disclosure
- D – Denial of service
- E – Elevation of privilege.

The following table summarizes the usual safety assumptions that may be invalid in the case of a cyberattack, the applicable attack methods according to the STRIDE classification, the system component or channel exposed to an attack and the prior countermeasures that should be implemented on the system to resist attack.

Object under attack	**Security/Safety assumptions**	**Defect or vulnerability exploited by attacker**	**STRIDE methods**	**Prior countermeasures**
System security control (input control)	Reasonable user behavior. Absence of cyberattack vectors which may cause physical damage.	Lack or inappropriateness of input validation. Bypassable input validation	TDE	Recheck assumptions about the user. Validation of the input control mechanisms according to domain area.
Monitoring sensors	Non-exposure of the monitoring sensors to cyberattacks.	Bypassable safety monitoring	STD	Recheck the physical protection and reliability of the sensors, implement tampering detection measures.
Channels transferring monitoring data	Non-exposure of the channels to cyberattacks.	Non-tamperproof monitoring	T	Recheck assumptions about access to the channels and the integrity of the data.

Safety enforcement mechanism	Non-exposure of the safety enforcement mechanism to cyberattacks.	Safety enforcement mechanism vulnerable and exposed to unauthorized access	DE	Verify the resistance of the safety enforcement mechanism to cyberattacks
Safety enforcement channel	Non-exposure of the safety enforcement channel to cyberattacks.	Safety enforcement mechanism vulnerable and exposed to unauthorized access	TD	Verify the resistance of the safety enforcement channels to tampering and denial of service.

Of course, not all the problems described above will take place on all systems. However, it would be worthwhile checking existing (possibly implicit) assumptions to ensure that the Security for Safety problem is properly addressed in every cyberphysical system.

Using this table, it is possible to refine the approach to threat modeling for the Security for Safety problem as follows. The appropriate attack tree or list of threats is created in the usual manner for every protection component listed in the first column. Then the threats are analyzed in accordance with the threat types listed in the fourth column and linked to potential safety issues. If there are links from the security threats to possible safety issues, special attention should be given to checking the explicitly or implicitly of the defined assumptions (second column of the table), as well as the design and implementation of the appropriate system component or channel (third and fifth columns).

The design and implementation of components, which are essential for the Security for Safety aspect, can be supported by MILS-based system architecture.

MILS architectural support of Security for Safety implementation

Here we show how the MILS concept can be used for hardening protection components on the architectural level within the scope of the Security for Safety problem. According to the order used in the table above, the following design principles are valid for these components:

1) It is worth implementing validation of untrusted external input in a separated MILS domain. This restricts the exposure of the whole system to cyberattacks. All other domains will receive input that was validated according to the given security constraints. This does not, however, remove the problem of possible incomplete checks of input data. For example, input files can be checked for viruses but not checked for their conformance to a specific format, while an invalid file format can still cause denial of service. Ideally, the outcome of an isolated security check must not cause any type of undesirable impact on either security or safety, but this requires the precise adjusting of security input control according to application-specific constraints.
2) Monitoring data (in the case of system-based monitoring or controlling system output) should be collected by sensors in the dedicated domains or by special means on the system kernel layer. From the point of view of non-exposure to cyberattacks, this is almost as effective as the fully separated system of external safety monitoring. Collecting the data from external sensors is also recommended in order to implement along with the agents

running in the detached domains. This reduces the possibility of tampering with data and protects the sensors (agents) from denial of service.

3) Monitoring data (in the case of system-based monitoring or controlling system output) should be transferred to the mechanism enforcing safety decisions using channels that are not exposed to application domains that may manipulate this data in a malicious manner.
   In particular, these channels can be provided by a separation kernel, which prevents data tampering.
4) The safety enforcement mechanism should not be externally exposed. The best way for embedded safety enforcement is to run it in a privileged MILS domain that is capable of taking control prior to any other mechanisms in the system.
5) Safety enforcement should use dedicated channel(s) to put the system or its components in a safe state and prevent damage. In the case of embedded safety enforcement, these channels must be provided by a separate kernel.

As shown, the Security for Safety problem can be effectively addressed on the basis of MILS architecture.

Conclusion

This work defines and explores the Security for Safety problem in cyberphysical systems. In order to understand the scope of this problem, the types of both information security issues and functional safety hazards and their relationships in cyberphysical systems were described. We determined the types of issues in cyberphysical system design and implementation that may cause safety problems in the event of cyberattacks.

The main sources of harm to safety caused by cyberattacks are improper assumptions about safety and security and defects and omissions in the implementation of appropriate mechanisms. We detailed the types of such implicit or explicit assumptions and described the nature of the defects. Based on these results, we proposed a unified approach to security threat modeling. This approach takes into consideration both informational threats and physical hazards which may be caused by potential cyberattacks. Finally, we outlined an MILS-based approach to the architectural design of system components to make them resistant to the safety issues that may arise from cyberattacks.

Bibliography

[1] Pietre-Cambacedes, Chaudet Claude «Disentangling the relations between safety and security», AIC'09 Proceedings of the 9th WSEAS international conference on Applied informatics and communications, 2009

[2] Giedre Sabaliauskaite and Aditya P. Mathur «Aligning Cyber-Physical System Safety and Security», Designing Smart Cities: Proceedings of the First Asia - Pacific Conference on Complex Systems Design & Management, CSD&M Asia 2014

[3] Fovino, I.N., Masera, M., Cian, A.D. “Integrating cyberattacks within fault trees”, Reliability Engineering and System Safety 94, 1394-1402, 2009

[4] Eames David Peter, Moffett J. The Integration of Safety and Security Requirements, Safecomp99, 27-29 Sept 1999, Toulouse, Franc.

[5] National Institute of Standards and Technology, Special Publication 800-82 R2 Guide to Industrial Control Systems (ICS) Security, 2015

[6] Langner, R., To Kill a Centrifuge. A Technical Analysis of What Stuxnet's Creators Tried to Achieve. Langner Blog, 2013, http://www.langner.com/en/wp-content/uploads/2013/11/To-kill-acentrifuge.pdf

[7] Federal Office for Information Security. The IT security in Germany 2014 (Bundesamt fur Sicherheit in der Informationstechnik. Die Lage der IT-Sicherheit in Deutschland 2014), https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/Lageberichte/Lagebericht2014.pdf?__blob=publicationFile/

[8] Miller C., Valasek C. Remote Exploitation of an Unaltered Passenger Vehicle. August 10, 2015, http://illmatics.com/Remote%20Car%20Hacking.pdf

[9] IEC 61508. International Electrotechnical Commission. International Standard IEC 61508: Functional Safety of Electrical/Electronic/Programmable Electronic Safety Related Systems

[10] Le Sueur G., Knobel P. Integrated Control and Safety: Assessing the Benefits, Weighing the Risks, http://oreo.schneider-electric.com/flipFlop/599765608/files/docs/all.pdf

[11] IEC 61511. International Electrotechnical Commission. International Standard IEC 61511: Functional safety - Safety instrumented systems for the process industry sector

[12] Microsoft. Microsoft Developer Network. The STRIDE Threat Model, https://msdn.microsoft.com/en-us/library/ee823878%28v=cs.20%29.aspx