Automotive Safety and Security Integration Challenges

Benjamin Glas[a], Carsten Gebauer[g], Jochen Hänger[h] Andreas Heyl[b], Jürgen Klarmann[c], Stefan Kriso[d], Priyamvadha Vembar[e], Philipp Wörz[f]

[dg]Bosch Center of Competence Functional Safety
[f]Chassis Systems Control, Engineering System Vehicle
[h]Corporate Sector Research and Advance Engineering
Robert Bosch GmbH
{carsten.gebauer, jochenulrich.haenger, stefan.kriso, philipp.woerz}@de.bosch.com

[b]Engineering Safety and Base System
Bosch Engineering GmbH
andreas.heyl@de.bosch.com

[ae]Bosch Center of Competence Security
[c]Software and Safety Consulting
ETAS GmbH
{benjamin.glas, juergen.klarmann, priyamvadha.vembar}@etas.com

Abstract: The ever increasing complexity of automotive vehicular systems, their connection to external networks, to the internet of things as well as their greater internal networking opens doors to hacking and malicious attacks. Security and privacy risks in modern automotive vehicular systems are well publicized by now. That violation of security could lead to safety violations – is a well-argued and accepted argument. The safety discipline has matured over decades, but the security discipline is much younger. There are arguments and rightfully so, that the security engineering process is similar to the functional safety engineering process (formalized by the norm ISO 26262) and that they could be laid side-by-side and could be performed together - but, by a different set of experts. There are moves to define a security engineering process along the lines of a functional safety engineering process for automotive vehicular systems. But, are these efforts at formalizing safety-security sufficient to produce safe and secure systems? When one sets out on this path with the idea of building safe and secure systems, one realizes that there are quite a few challenges, contradictions, dissimilarities, concerns to be addressed before safe and secure systems started coming out of production lines. The effort of this paper is to bring some such challenge areas to the notice of the community and to suggest a way forward.

Note

- The term “Functional Safety” relates to ISO 26262
- The term “Security” is used to mean Automotive Embedded Information Security
- All examples used in this paper are fictitious and do not necessarily reflect either concrete requirements or solutions.

1 Introduction

There have been calls in the past from the academic community for safety and security communities to work together [Avi04]. The need for building safety and security in automotive vehicular systems is real today, as we have started building safety-relevant systems with security requirements.

The impact of security on safety has been widely discussed and a consensus seems to have emerged. [Bur12] makes a case for the consideration of functional safety hazards arising out of malicious manipulation in addition to safety hazards arising out of systematic failures and random hardware failures by the ISO 26262 standard. [Bur12] as well as [Cze13] describe a functional safety and security engineering process. [Bur12] discusses identifying security goals in addition to safety goals in a combined functional safety-security analysis. That, security is a pre-requisite for safety and that safety could be the driver for security is an established view. There is ‘that much’ forward movement on this subject.

But is this ‘much’ really much? On the other side, there are some dissimilarities between the safety and security disciplines. The maturity-level of these disciplines is different. Automotive security is a younger discipline when compared to automotive safety. Safety has matured over decades and has had successful standardization efforts. Safety can closely align itself to legal requirements from product liability. The differing maturity levels, grey areas in law, dissimilarities in content create real challenges to realize safety and security together. The focus of this paper is to discuss these challenges and to suggest solutions or solution directions.

2 The Challenge Areas

Functional Safety discipline considers systematic and random hardware failures as hazard sources. Security considers a malicious and intelligent adversary as a threat source in addition to natural disasters and systematic failures. The unacceptable consequences for safety are loss of human life and injuries. Security as a discipline has a broader range of unacceptable consequences: human life, human security, loss of reputation, financial losses, legal violations, loss of intellectual property, damage critical infrastructure etc. These differences could be reconciled. Though, security threat sources and threats could be varied, for our purposes, we could consider ‘that’ subset of security threats that lead to safety consequences. But, there are many other differences that cannot be this easily reconciled. We focus on such challenge areas in this paper.

2.1 The Misuse Challenge

[Bur12] discusses a method for performing a joint functional safety and security analysis and illustrates the same with an example. The security analysis process begins by identifying 'Misuse cases'. The misuse cases are later analyzed for safety risks. The differences begin right here. The differences become apparent when safety and security teams discuss what 'misuse' means to each of them. The safety community believes in preventing 'foreseeable misuses' by the 'user' of the product. But, when the security community talks about misuse, they are talking about a whole range of 'negative scenarios' that could be brought about, not just by the user (who could also misuse the system, for example, by performing chip-tuning), but also by an external malicious agent who possesses the wherewithal in terms of capabilities and resources to cause not just loss of life and injuries but a greater large-scale damage. The picture below illustrates this range of possibilities.

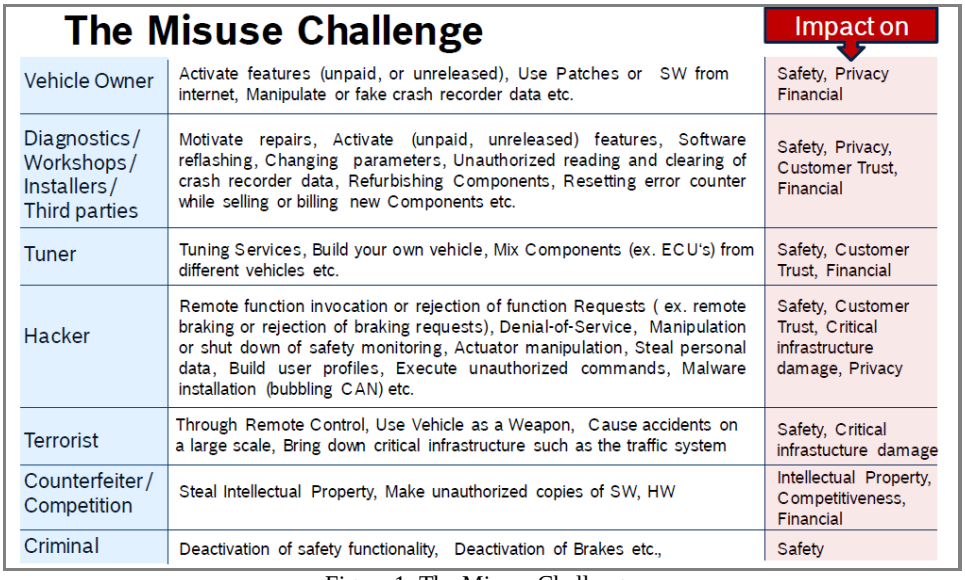

Figure 1: The Misuse Challenge

These acts would perhaps qualify for a carefully hatched 'sabotage'; not a foreseeable misuse really, and hence would amount to no penalties from the product liability legislation for non-consideration of the threat in product development. What sort of security 'Misuse' would be a misuse that must be prevented, on which efforts need to be invested? Do we want to make distinctions between acts that could be easily performed by a 15-year old armed with a computer, a set of downloaded tools and an instruction manual from the ubiquitous 'You Tube'? Do we accept the risks presented by a well-versed hacker, a cyber-criminal or a terrorist as sabotage that need not be prevented or detected? Where should the line for risk acceptability be drawn? What is the legal opinion on this? Do we need consensus and policies here, at least, for the automotive world?

Our view on the subject is as follows: The EU directive 2001/95/EG, Article 2 gives the following definition: "'safe product' shall mean any product which, under normal or

reasonably foreseeable conditions of use [...] does not present any risk or only the minimum risks compatible with the product's use, considered to be acceptable and consistent with a high level of protection for the safety and health of persons [...]" [EG01]. Safety activities focus on normal and reasonably foreseeable conditions of use. In the German product safety act ("Produktsicherheitsgesetz") this is named "beabsichtigte und vorhersehbare Verwendung" ("intended and foreseeable conditions of use") [PSG11]. This is also the intention of ISO 26262: When performing a hazard analysis and risk assessment (H&R), intended use (normal conditions of use) inclusive reasonably foreseeable use is considered (ISO 26262-3, 7.4.3.7, NOTE 2). For example, when performing a H&R it is "intended use" that someone does not drive faster than a given speed limit, e.g. 50 km/h. But it is reasonably foreseeable that someone drives "a little bit faster" than this given limit, e.g. 60 km/h. But it is not reasonably foreseeable that he drives much faster, e.g. 200 km/h. But it is difficult to define a clear borderline between "foreseeable" and "not foreseeable" here.

According to ISO26262 safety is defined as the "absence of unreasonable risk" (ISO 26262-1, 1.103). The risk coming from a system is the "combination of the probability of occurrence of harm and the severity of that harm" (ISO 26262-1, 1.99). Possible sources of risks are malfunctions coming from random hardware faults during operation of the system as well as systematic faults during development of the system. Both these kinds of faults are addressed in ISO 26262, "functional safety of road vehicles".

Another source of risk could be a malfunction coming from the manipulation of the system, e.g. a hacker attack, which could provoke safety-critical behaviour of the vehicle (independent thereof whether this is intended by the hacker or if it is only an unintended side effect). If the probability of such a manipulation (in combination with its severity) is so high that it leads to a "unreasonable risk" level, measures for risk reduction are necessary. Basically, measures against manipulation are addressed in the area of "automotive security", not in the area of "functional safety" – but they have to be considered as "safety measures" in the sense of "risk reduction measures". For example, if an engine management system is not hardened against hacker attacks, such an attack could lead to safety-critical behaviour of the vehicle (e.g. braking system not being able to deal with a higher engine power).

Even here, it is difficult to define clearly the probability of manipulations and with that, the necessary risk reduction measures. Additionally, this cannot be expected to remain static over time: the probability is expected to rise, which would lead to an increase of the risk level from "reasonable" towards "unreasonable", which would make (more) risk reduction measures necessary. It is the task of field monitoring process to observe this development. If automotive security is a measure for risk reduction to ensure safety, this leads to the necessity to coordinate the safety and security activities.

2.2 The Risk Assessment Challenge

Going further, [Bur12] presents a way of performing a risk analysis. They present an approach in which safety risks are separately assessed and security risks are separately assessed using their individual methods, but as a part of H&R. But, does this kind of risk

analysis produce the right picture? The picture below displays the safety risk model along with how an attacker could fit himself into it.

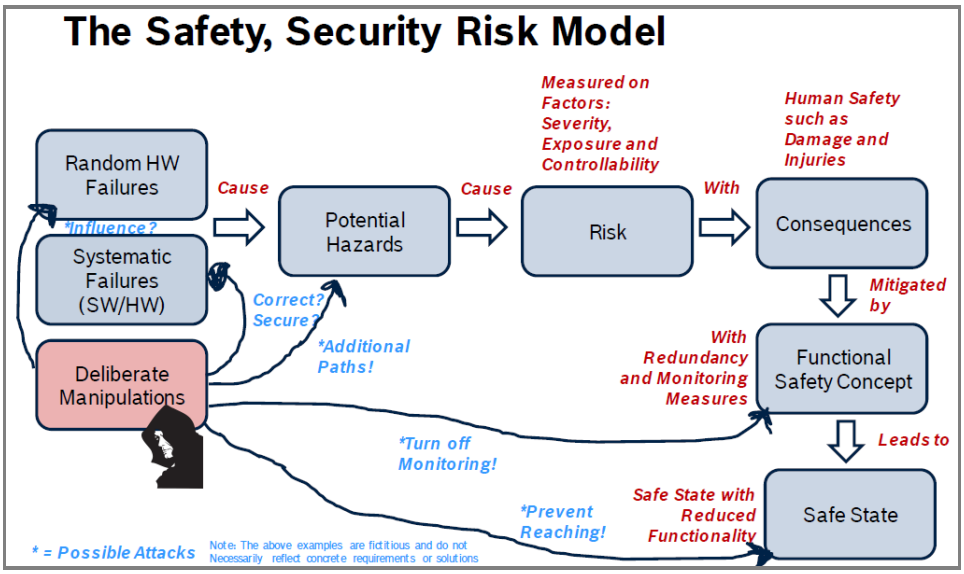

Figure 2: The Safety, Security Risk Model

The safety risk model is oriented towards dealing with risks that arise due to natural or random causes in nature. But, the introduction of an 'intelligent' adversary alters the risk landscape significantly. This makes the 'Hazard and Risk Analysis (H&R)' on the safety side and 'Threat and Risk Analysis(T&R)' on the security side, inherently incompatible.

The H&R tries to find and classify the risk, which is related to an item. The risk classification is documented via the "ASIL" (Automotive Safety Integrity Level). The classification is based on three parameters: (a) The potential severity of an hazardous event (severity), (b) probability of an operational situation (exposure) in which this event might happen, and the (c) ability of traffic participants to control the event (controllability). Within the classification, external safety measures can be considered to be affecting the exposure, severity or controllability to reduce the ASIL.

As a result of the H&R, safety goals with ASIL shall be defined. If similar safety goals are determined, these may be combined into one safety goal. In this case, the highest ASIL shall be assigned to the combined safety goal.

The picture below presents a high-level comparision of the safety and security risk models.

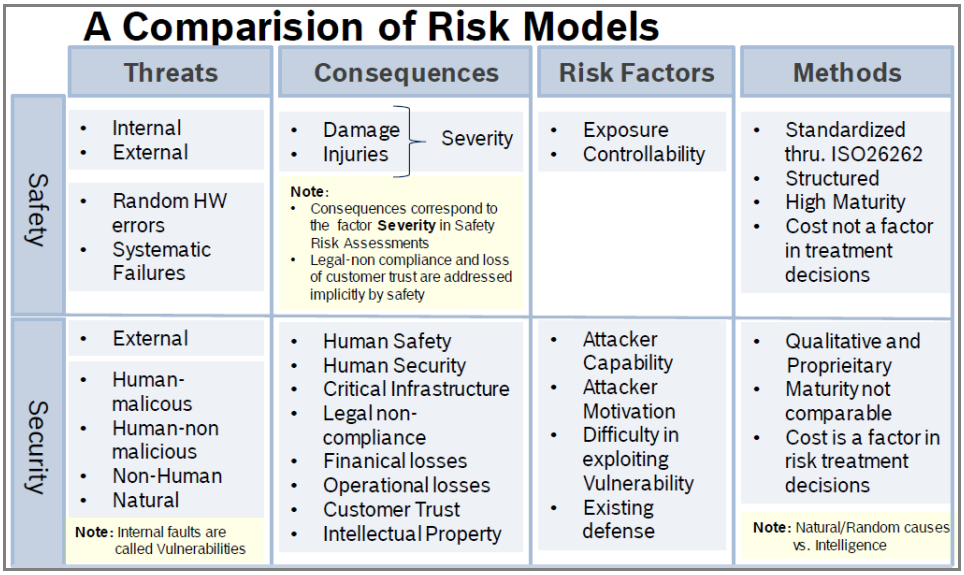

Figure 3: A Comparision of Safety and Security Risk Models

"Consequence" on the Security side, and "Severity" on the safety side is the common parameter for security and safety. The probability-related parameters nevertheless are quite different, adressing the attack potential of the hacker and the attack potential the system is able to withstand: available time for an attack, required specialist expertise, required system knowledge, window of opportunity to access the target of attack, required equipment etc.

Safety is a common asset for both the domains. Both try to assess the safety-related risk which might be a result of random or systematic faults on the one side or deliberate manipulation on the other, to derive the required "strength" of the corresponding countermeasures. Both get to the risk classification by applying the mathematical definition: risk = severity X the probability of an event. Therefore, it seems feasible to base ones T&R on the results of the H&R to save time [Bur12, Cze13]. In the following chapter, we show by counter-arguments why this could lead to misunderstandings and why the commonalities seem to be limited to the rough process steps and to the basic definitions.

The first point of criticism adresses the classification parameters, which can not be used on both sides. For example one can construct many E1-situations, in which a malfunction is hard to control (C2 or C3) and leads to severe or lethal injuries (S2 or S3). But due to the fact, that these situations are so rare, ISO 26262 will classify the corresponding malfunction with a low ASIL, e.g. QM or ASIL A. For a hacker this small likelihood is not that relevant, since he can either wait for such a situation or even provoke it before triggering the corresponding attack. At an advanced level, he can even have a "sleeping" function waiting for a specific situation by evaluating vehicle data and triggering an attack automatically.

Additionally, the hacker might be able to also influence the controllability. This can be achieved by performing a parallel attack to distract the driver in the moment of the main attack, e.g. by impairing his field of vision by flushing wiper fluid [Kos10] or irritating him by increasing the radio volume.

Composite attacks on available and considered external measures (e.g. ESP being external of the item "powertrain") could also be used to impair their capability to mitigate a hazardous malfunction of the system. Therefore the focus of the H&R on an item seems to be insufficient for theT&R, which has to widen its scope to the whole vehicle or even the environment.

The conclusion is, that the "probabilistic" filter of the H&R makes it rather impossible to come up with a simple mapping to transfer the results of a H&R into a T&R. Rather, all situations and parameters have to be assessed again under a security point of view ("unlikely vs. unattractive"), applying security-specific methods and checklists. Since, within the H&R, typically implicit assumptions are made, which might lead to a reduction or grouping of situations, even the situation catalogue resulting out of a H&R might be incomplete for the security considerations in the T&R. Probably only the definitions of the terms and the various classes of "severity" and "controllability", that are already defined in ISO 26262, could be a candidate for a common use.

But even if these methods (H&R and T&R) don't go together very well, the experts of these two fields might and should. Hence it is recommended, that experts of both fields exchange and discuss their risk analysis results: (a) to check them for completeness, at least regarding the asset "safety", (b) to become aware of the subsequent functional and technical requirements for each domain, that might affect each other later on in development, and (c) to gain a mutual understanding of each others field and their special needs, approaches and contradictions.

2.3 The Solution Space Challenges

The issues deepen as we move into the area of finding solutions that are acceptable to both the safety and the security communities.

Safety and security share the common goal of protecting integrity – meaning, the correct and intended functionality of the system against failures, errors, and mainly external influences. Security usually has more goals and other assets to protect, but in this area one finds the greatest overlap. Towards meeting this objective, both use a similar set of mechanisms, methods and resources, e.g. adding redundancy, monitoring, testing, and verification. Naturally, if applied independently, this often results in conflicts since both disciplines need access to the same set of limited resources: bandwidth of communication, storage space, processing time, access to flash and RAM, and others. Before we examine approaches to generate synergies, let us consider two examples of actual or possible conflicts.

2.3.1 Example 1: Failure Analysis of Issues during Product Usage

Field monitoring of a product is required by several laws (e.g. w.r.t product liability), standards (e.g. ISO 26262-7), as well as internal and external regulations.

Typical points of analysis from a safety point of view include:

- Flash analysis by identification of software version, checking the integrity of flash hardware and content and even dumping the flash content
- RAM analysis by checking hardware and content, and also modifying RAM content during runtime for debug reasons
- Check of EEPROM content (e.g. failure memory)
  Check of ECC (error correcting code) information of microcontroller registers

In general, these analyses require access to the microcontroller in debug mode.
Typical stakeholders for these activities are

- Tier 1 Developer (System, HW and SW)
- Tier 1 Manufacturing and Quality Assurance
- OEM Manufacturing and Quality Assurance

From security point of view, this is at least partially a contradiction to the security goals and interests, because the security interests would read like the following:

- Protect flash content against manipulation and re-engineering (protect intellectual property)
- Protect RAM content against manipulation, restrict access during runtime (in debug mode) used for system analysis to authenticated and authorized users
- Protect EEPROM content to prevent misuse cases like:
  - modification of failure entries in order to shift responsibility for an event (from driver to system)
  - read access to confidential data, for example, as a part of re-engineering
  - read access to user specific data, which might violate privacy rights of the user.
  - write access to system configuration data to activate unpaid functionalities (which could lead to loss of revenues)

It can be seen that, while the basic goal of preserving ‘Integrity’ remains the same across the disciplines, the methods employed by safety and security may conflict in the actual application. One good example is the ease of access to system resources. While safety implicitly expects that access should be easily possible for a quick and easy analysis, security would restrict access as strictly as possible via authentication and authorization mechanisms. In many cases, this can be resolved by granting authorization specifically to valid safety-monitoring entities at the cost of added (management) overhead. Nevertheless a trade-off needs to be found between allowing access for legitimate and authorized safety features while preventing an attacker from using this access to violate security goals. In order to distinguish a valid safety-mechanism from an attack, a co-design for safety and security mechanisms is necessary with need for compromise on both sides. Some safety mechanisms need re-evaluation, as to whether they can be used in presence of security needs.

2.3.2 Example 2: RAM Test

As part of a standard safety concept, RAM tests are required. Usually, these RAM tests include (temporary) modifications of RAM content by writing test patterns into selected RAM sections. For other users of the RAM such as bus masters, it is impossible to distinguish between real RAM data and test patterns. This could lead to unintended or even safety-critical behavior of the system. In order to prevent usage of these RAM test patterns, usually all bus masters are stalled during the RAM test execution. This is a well-established design pattern from the safety perspective.

But, from a security point of view, one of the main security requirements is to prevent stalling of main security (hardware) components like Hardware Security Modules (HSM). Without further measures this would lead to the possible misinterpretation of RAM test patterns as valid code or data by HSM and therefore random HSM behaviour.

Both examples do not show fundamental contradiction in requirements. Nevertheless, they show that :

- Current well-established design patterns are not feasible anymore when security requirements are added
- Mutual dependencies have to be analyzed
- Common design for safety and security aspects is necessary in future

Current automotive reality is in the situation, that safety is an integral element of system design, while security features are new to many automotive subsystems. Integration of both sets of requirements needs compromises or trade-offs on both sides. An understanding on the side of security that the established safety mechanisms are not easy to change, and an acceptance on the side of safety that security is not just a “plug-in” that can be applied independently and on top of the legacy safety system, but needs integration and adaptation to the existing and proven safety approaches.

3 Synergies between Safety and Security

3.1 The CRC and the MAC

One area where safety and security goals coincide is in the area of integrity protection for data – both in transit and in storage. In order not to use corrupt or manipulated data, integrity has to be protected or at least loss of integrity has to be detected. The difference between the domains is the perceived threat for the integrity. While (functional) safety aims at protection against systematic errors and random errors caused by malfunction or unintended interference, security additionally wants to protect against targeted, intended and possibly malicious manipulation. This leads to different approaches for protection, which nevertheless have some common properties. The general idea is always to add redundancy to the data, so that a consistency check detects deviation from the original data. The amount (number of bits) of redundancy correlates with the probability of detecting an error, the more redundancy invested, the more errors can be detected.

Differences are in the approaches to generate this redundancy. In the following section, we take a look at two prominent representatives.

Safety: The Cyclic Redundancy Check (CRC)

Cyclic redundancy check denotes a class of mechanisms to detect and partly also to correct randomly distributed errors using additional redundant data generated by binary polynomial division. The polynomial can be chosen to adapt to different requirements. Basic assumption is usually a binomial distribution of single bit errors, which implies e.g. that the probability of an error pattern decreases with the number of changed bits and particularly single bit errors are more likely than multi bit errors. Most CRCs in use guarantee detection of errors of some classes (e.g. single bit errors or all errors up to a certain number of changed bits) and even offer correction for a subset of these classes. CRCs are easy to generate and evaluate both in hardware and software and need no secret information to be computed. They are widely used, e.g. in automotive communication (e.g. CAN bus protocol).

Security: The Message Authentication Codes (MACs)

Message Authentication Codes are a class of mechanisms using basic cryptographic primitives like keyed cryptographic hash functions or block ciphers to create integrity tags that can only be created and evaluated with the knowledge of a secret key. They belong to the cryptographic class of symmetric primitives, meaning that the key used for generation of a MAC is the same as the one needed for verification.

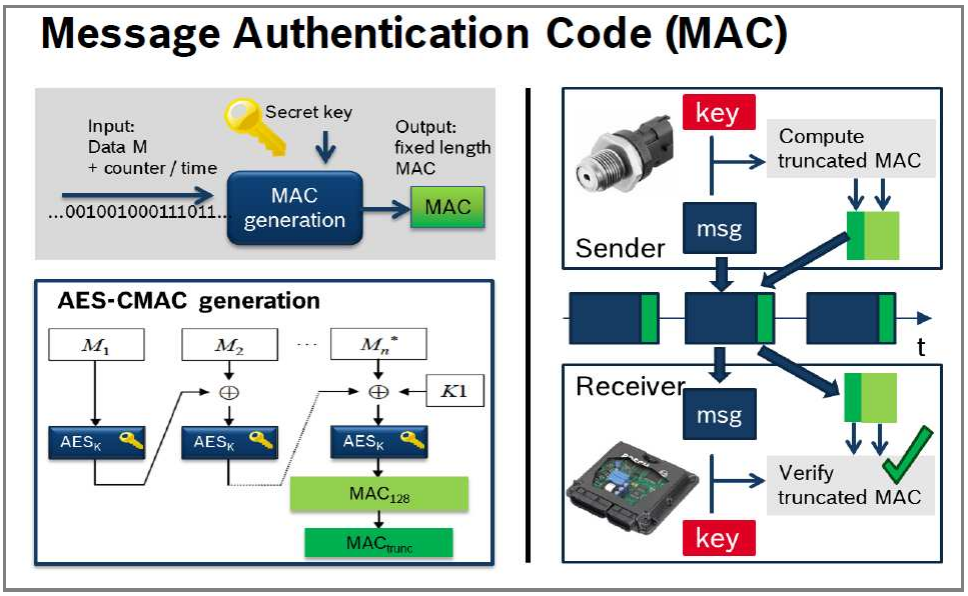

Figure 4: The Message Authentication Code(MAC), an Introduction

In contrast to CRCs, MACs have different basic design criteria, since the basic goal is authentication of data and/or originator of the message. This leads to the desired property, that is has to be infeasible for an adversary to create a valid MAC for a

message without knowing the key. This has to be guaranteed even if the abstract adversary has access to an arbitrary number of message-MAC-pairs. So, even if for any number of similar messages the MAC is known, “derivation” of the MAC tag for a different message must not be feasible. This also leads to the properties, that for similar messages, the MACs must not be similar, at least statistically, and that from a MAC it should be infeasible to extract information about the message.

Looking at cryptographic MACs, it can therefore be usually assumed, that errors/manipulations in the message are detected with a constant probability of $2^{-(\text{length of MAC})}$ for all errors, independent of the error class (e.g. hamming distance). No guarantees can be given for specific types of errors. Also no correction is possible. While mathematically this is the best error detection possible with the redundancy invested assuming a uniform distribution of errors, this is not the best detection assuming a binomial error distribution.

Algorithmically, the receiver cannot distinguish between a random error and a malicious manipulation of data. Therefore the MAC aims at detecting any deviation from the original message which includes and goes beyond all error causes looked at from a safety point of view. Therefore a MAC could in principle replace a CRC for error detection purposes in resource constrained in systems in order to enhance the functionality to authenticity protection, if the uniform detection probability is acceptable from a safety point of view.

3.2 Virtualization for Embedded Systems

Virtualization in computing refers to the act of creating virtual computers on one physical computer. Here, we refer to such virtualization technologies only, where the hypervisor runs directly on the hardware. These are most accepted for embedded systems and provide features that support both safety and security.

The main features of virtualization that are of interest to us are:

- Strong isolation between virtual computers: Two virtual computers do not know one another and run like independent physical computers. This property is provided by a hypervisor which provides a strong separation between both virtual computers
- Communication: A communication channel between both virtual computers may be provided and controlled by the hypervisor (comparable to a data bus between physical computers).
- Hypervisors for embedded systems are typically very small and comprise only some thousand lines of code. Experts say that they come with a “small trusted code base”.
- Cost benefit: Introduction of virtual computers does not increase serial production costs for physical computer and do not increase further costs in the vehicle e.g. for harness or for mounting.

The picture depicts hypervisor uses for safety, security and both safety and security.

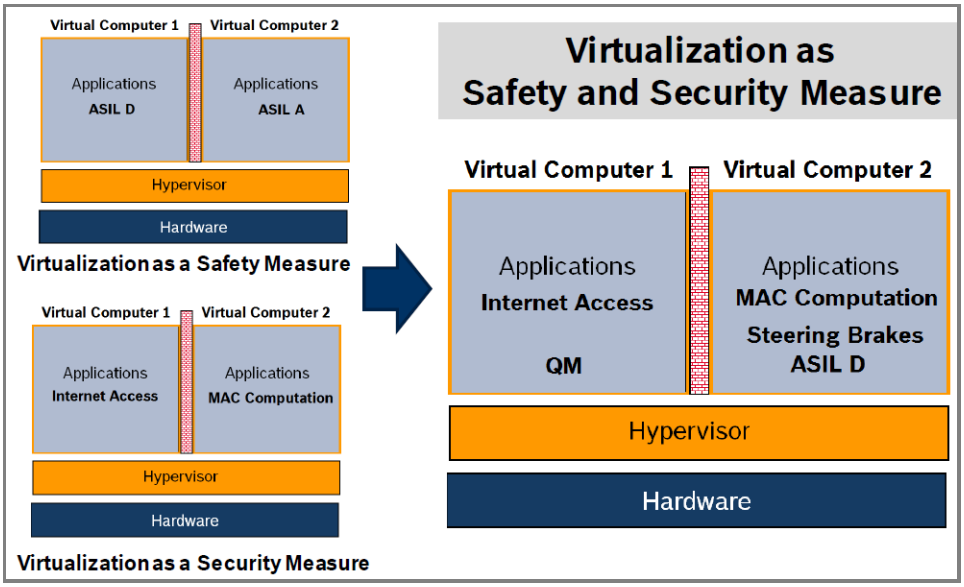

Figure 5:Virtualization as Safety and Security Measure

Virtualization as a Safety Measure: The aim of functional safety standardizations like ISO 26262 is to minimize hazards and risks to a reasonably low level. As hazards and risks are of different nature, it is customary to encounter different criticality levels (ASIL A to ASIL D) during system design. ISO26262 demands development according highest ASIL for whole software as long as sufficient freedom from interference cannot be guaranteed between applications with different ASIL. Development according to highest ASIL is often not possible when software from different sources shall run on one physical computer. The strong isolation feature of virtualization guarantees required freedom from interferences between applications of different ASIL. Thus, each virtual computer may run software with an ASIL of its own.

Virtualization as a Security Measure: Virtualization comes with features which are a great help for designing secure systems. Due to strong separation of virtual computers, each one may run independent of the others comparable to applications running on individual physical computers. Thus a security attack on one virtual computer does not affect the others. In addition, each virtual computer may be stopped or restarted independently. That allows for a design where applications prone to security attacks (e.g. applications with internet connectivity) run in one virtual computer while applications with safety or security critical tasks (e.g. steering brakes or MAC computation) run on another virtual computer. Such a design is further supported by the small trusted code base that hypervisors typically come with. Typically, small code of the hypervisor can get hardened to ensure the strong isolation between the virtual computers.

Virtualization as Safety and Security Measure: Virtualization may serve as a measure for both safety and security on one physical computer as shown in the figure above. The figure shows one physical computer on which two virtual computers run applications of different ASIL and different security criticality.

3.3 Making Programs Run Time Error Free

A run-time error is an error detected after or during the execution of a program in contrast to the errors detected during compile-time. Run-time errors such as invalid usage of pointers and arrays, invalid ranges and overflows, division by zero, invalid function calls (wrong parameters, recursion), or use of uninitialized variables are often hard to find by running test cases. Run-time errors may lead to arbitrary failures.

Run-time errors are classified as systematic faults in safety and may lead to safety relevant malfunctions. ISO 26262 (book 6, table 1) suggests the use of language subsets to exclude language constructs which could result in unhandled runtime errors. However, in the most used embedded programming language C, runtime errors can be caused by *basic operations and thus absence cannot be ensured by exclusion by a language subset.*

According [Wal09.] most security attacks are input or output related. Today's security vulnerabilities lie rarely in cryptographic algorithms or protocols. Almost always these security vulnerabilities are implementation related. Lists like the "Recent Vulnerability Notes" [Cer40] demonstrate that very well.

Thus, the guarantee of absence of runtime errors helps safety and security significantly.

Sound abstract interpretation may guarantee absence of all runtime errors. The aim of abstract interpretation is to obtain information about the behavior of programs, by abstracting parts of the program and the instruction retraces step by step. In abstract interpretation one focuses on some aspects of the execution of the instructions, abstracting out some details, giving an approximation of the program semantics, which is sufficient for the desired purpose. "Sound" means all runtime errors are found. Thus we don't refer to methods and tools here which are unsound, which means, they may find only a portion of all runtime errors. Abstract interpretation consists of considering *abstract semantics*, which is a superset of the concrete program semantics. The abstract semantics cover all possible cases. If the abstract semantics are ok then, so also the concrete semantics [Abs14]. Sound abstract interpretation is able to detect all kinds of runtime errors in C source code.

3.4 Combining Virtualization and Runtime Error Removal

The usage of the hypervisor in the approach depicted in the previous chapters is based on the ability to create a strong separation between the virtual computers. The strong separation is realized by a typically small trusted code base. But even that small trusted code base may contain runtime errors which may endanger safety and security goals.

Hardening of hypervisor code by sound abstract interpretation is a clear possibility.The application of sound abstract interpretation for a hypervisor has the advantage of being cost-efficient due to its small code base. Together, these measures offer a great potential in being used in applications requiring both safety and security.

4 The Misconceptions

Until now, we discussed challenges and synergies. There also exist misconceptions that arise from carrying the 'Safety Paradigm' into Security. A prominent example of this is with regard to the 'Safe State'. Such questions are asked: Upon detection of a hazardous situation, safe state would be to 'Open the doors' to let the passengers out, but would 'Secure State' be to shut the doors to keep the intruders out? It would perhaps be necessary (in hypothetical situations) for a safety-relevant system to write to some areas of memory, modify the internal state to bring the system to safe state in emergency situations. Is security against this? Will it prevent it? Are the requirements contradictory?

The answer is: There is no such thing called a 'Secure State' defined in the domain of security. Through information security, we intend to protect information assets from loss of confidentiality, integrity and availability against an unauthorized intruder. Taking this idea, a step further, even if there were to be a 'Secure State', it would be an integral 'Safe State' in which the confidentiality of sensitive information is appropriately protected and the resources needed to attain a safe state and the resources needed to persist in the safe state are available and are protected from malicious denial-of-service.

In general, security takes on those goals that are expected of it by the system. Hence what security does is dependent upon the goals of the system in question. Security per se, does not possess independent goals of its own. This holds for any safety-relevant system.

5 Methodological Proximity

There is methodological proximity between the two disciplines. At a high level of abstraction, the engineering processes appear similar. But, there is more to the story.

The risk models of safety and security are entirely different in nature. Safety Risk Model is based on random events in nature, whereas, security considers risks due to malicious human intelligence. Hence the risk analysis, assessment approaches differ. The risk models cannot be superimposed on one another, nor can there be uniform approaches for assessment though both protect the same asset 'safety'. Hence, Safety and Security require a Risk Co-Analysis approach.

At the level of countermeasures, safety takes a control system view and a detective approach, but the focus of security would still remain on preventative measures (though detective and recovery measures do have their place). There are challenges and potential synergies in safety and security measures, that require a shift in the thinking on the

safety side. For the security side, a word of caution is in order. Projects with safety relevance are not green-field projects. There is work in integration which needs time and interaction. Hence, we propose safety+security Co-Design to achieve the integration.

At lower levels of abstraction, such as coding, security needs to go a step further from safety. Safety needs correctness, but security needs correctness and some thing more to be secure.

In the area of verification and validation, security test cases can build upon existing test-case development methods for software testing such as equivalence partitioning, boundary value analysis, multiple condition coverage etc., with additional test cases for testing the preservation of Confidentiality, Integrity and Availability (CIA) at all levels.

6 Conclusion

Safety-Security integration is an important integration for automotive vehicular systems. The processes look similar and could be performed together. But, attention to detail is essential. The safety and security communities need to engage at a deeper level than today to make the integration happen. Any process for safety and security (isolated or combined) needs to be aware of the inherent differences and commonalities between the two disciplines. A security process needs to be 'Safety Aware' and vice-a-versa. Hence, safety and security processes performed by different experts - is clearly not the way forward. Separate safety and security standardizations by themselves don't help either. Finding vocabulary, developing tools that could be used by both communities in a similar manner, resolving or at least developing similar understanding of grey areas in law, looking for mechanisms, architectural building blocks, design patterns that could work for both the domains holds the key to bringing forth a meaningful, working and 'Safe' integration

References

[Avi04] Avizienis A, Laprie J-C, Randell B, Lanwehr C, "Basic Concepts and taxonomy of Dependable and Secure Computing", IEEE Transactions on independent and Secure Computing". January-March 2004.

[Bur12] Simon Burton, Juergen Likkei, Priyamvadha Vembar, Marko Wolf, "Automotive Functional Safety = Safety + Security", In: 1st International Conference on Security of Internet of Things (SecurIT 2012), Kerala, India

[Cze13] Czerny, B., "System Security and System Safety Engineering: Differences and Similarities and a System Security Engineering Process Based on the ISO 26262 Process Framework," *SAE Int. J. Passeng. Cars – Electron. Electr.Syst.* 6(1):2013, doi:10.4271/2013-01-1419

[EG01] Directive 2001/95/EC of the European parliament and of the council of 3rd December 2001 on general product safety

[PSG11] Act on making products available on the market (Product Safety Act) / Gesetz über die Bereitstellung von Produkten auf dem Markt (Produktsicherheitsgesetz, ProdSG), 8. November 2011 (BGBl. I S. 2178, 2179; 2012 I S. 131)

[Kos10] Karl Koscher et al. “Experimental Security Analysis of a Modern Automobile”, University of Washington Seattle, University of California San Diego, http://www.autosec.org/pubs/cars-oakland2010.pdf, 2010

[Wal09.] Walter Kriha, Roland Schmitz, “Sichere Systeme, Konzepte, Architekturen und Frameworks”, Springer Verlag, 2009

[Cer14] Vulnerability Notes Database, http://www.kb.cert.org/vuls/, 29.10.2014

[Abs14]“Abstract Interpretation in a Nutshell”, http://www.di.ens.fr/~cousot/AI/IntroAbsInt.html , 29.10.2014

[ISO11] International Organization for Standardization, ISO 26262, “Road Vehicles – Functional Safety”, International Standard