Safety and Security Concept for Software Updates on Mixed-criticality Systems

Imanol Mugarza*, Irune Yarza*, Irune Agirre*, Fabrizio Lussiana†, Stefania Botta†
*Ikerlan Technology Research Centre, Basque Research and Technology Alliance (BRTA), Spain; †Marelli, Italy
{imugarza, iyarza, iagirre}@ikerlan.es; {fabrizio.lussiana, stefania.botta}@marelli.com

***Abstract*—The raising connectivity of critical embedded systems makes them vulnerable to cyber-security attacks that compromise not only privacy but also safety. This results in intricate dependencies between functional safety and security, and higher demands to address both disciplines simultaneously. However, there are still many gaps on the common application of functional safety and cyber-security standards. Over-The-Air (OTA) software updates are a clear example of this challenge. While the installation of regular software upgrades is a crucial cyber-security practice to keep the system up-to-date with the latest security patches, they might involve high re-certification efforts and costs from a safety standpoint. In this paper, a safety and security concept for software updates on mixed-criticality systems is presented. Particularly, a combined safety and security risk assessment on an automotive use case is performed and risk mitigation measures proposed.**

***Index Terms*—software updates, safety, security, concept, risk assessment, risk treatment, mixed-criticality**

I. Introduction

The rapid evolution of hardware and software in Mixed-Criticality Cyber-Physical Systems (MCCPS) has surpassed the capabilities of current safety- and security-oriented design methodologies. Generally, standards used in the certification process of such systems reflect the state of practice in industry rather than the state of art. As a result, they do not evolve as fast as technology, and they do not provide explicit guidance for next generation architectures yet [1].

Over-the air (OTA) updates are a clear example of this trend, a technology commonly used in consumer electronics market that is now being adopted by critical industries such as the automotive [2], [3]. The benefits of over-the-air updates improve maintenance (e.g., bug fixing, security patching) and give enhanced flexibility to the systems, making it a key technology to stay competitive in the market. However, software updates or modifications in general have a very different treatment and relevance on the safety and security-critical domains [4]. This difference is highly motivated by the asymmetric impact that in-service experience has in both domains [5], [6]:

- In the safety domain, product operation hours and history, together with field failure data, are key indicators to gain evidence on the absence of systematic design faults in a product. As a result, confidence on a system increases with its time in service.
- In the security domain, on the contrary, new security flaws and weaknesses are disclosed every day and the security trust level decreases over time.

As a consequence, software updates are a required practice according to security standards in order to regularly solve new security vulnerabilities. On the contrary, modifications on safety-critical systems are discouraged and usually limited to unavoidable maintenance activities like solving faults that resulted in incidents or adapting to new or amended safety legislation (e.g., IEC 61508 [7]) and they might involve high re-certification efforts and costs [4], [6]. In addition, the increased connectivity of critical systems result in intricate dependencies between safety and security and security threats and vulnerabilities could jeopardize functional safety. For all these reasons it is increasingly important to simultaneously address safety and security needs from early design stages.

The UP2DATE European project [8], [9] seeks to address the main dependability challenges brought by OTA updates to the critical domain, with special focus on safety, security, availability, maintainability, and the increasing platform complexity of emerging heterogeneous Multiprocessor System on a Chip (MPSoC) devices. This paper presents a safety and security concept of a mixed-criticality software updates enabled system, based on the common application of IEC 61508 [7] and IEC 62443 [10] for functional safety and cyber-security respectively. To this end, a combined safety security risk assessment methodology is presented. This methodology is then applied to the UP2DATE architecture, a mixed-criticality system enabling OTA updates, presented in [8], [9] and, as a result, safety and security risks are identified. Finally, the safety and security countermeasures that shall be applied to reduce system risks are defined. All this process is followed based on a next generation automotive use-case that combines advanced high-performance functionality with critical functions.

This paper is organized as follows: after this introduction, the employed safety and security risk assessment and treatment methodology is presented. After that, the UP2DATE architecture is described and the system concept specified. Following, the safety and security risk assessment is provided. Lastly, related work is presented and conclusions drawn.

II. Methodology

For the systematic safety and security risk assessment, the well-known ISO 31000 [11] and ISO 27005 [12] standards are considered. This process is aligned with the risk assessment method described by ISO/SAE 21434 [13], which also references ISO 31000. It should be pointed out that the IEC 62443

[10] standard also recommends (among others) the ISO 27005 as basis for risk identification and assessment. Figure 1 shows the followed high-level safety and security risk assessment methodology.

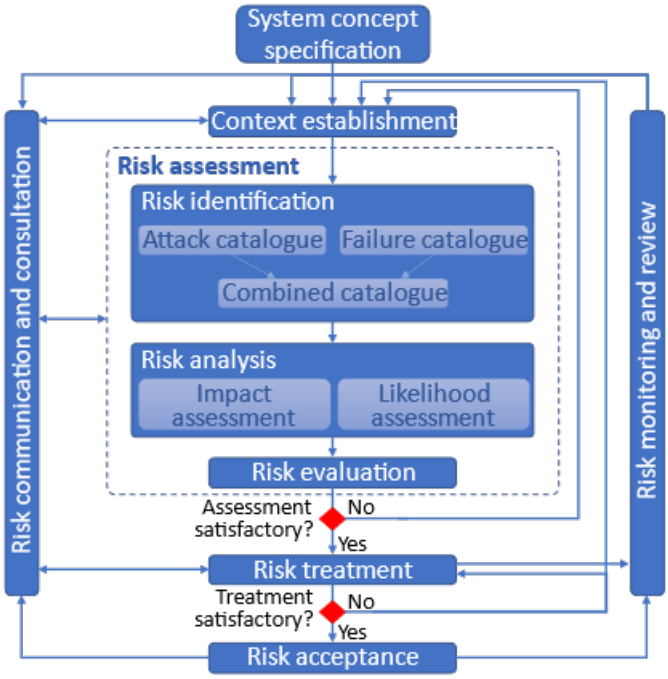


Fig. 1: Safety and security risk assessment methodology.

Besides, for the detailed risk analysis, the MAGERIT [14] (version 2) risk analysis and management methodology, elaborated by the Spanish National Cryptologic Centre (in Spanish "Centro Criptológico Nacional, CNI") is used. This methodology, which extends and tailors the requirements and processes of ISO 27005 [12], is endorsed and recommended by both national and international cybersecurity agencies, such as the European Union Agency for Cybersecurity (ENISA) and INCIBE (in Spanish "Instituto Nacional de Ciberseguridad"). Table I shows the employed threat catalogue.

III. System Concept Specification

Critical system development processes always starts with the the system concept specification aligned with the lifecycles dictated by standards. This section summarizes the UP2DATE architecture [9] that supports safe and secure software updates for both intelligent and resource intensive mixed-criticality systems as well as for legacy control devices. To this end, UP2DATE architecture is characterized by the inclusion of a high-performance mixed-criticality gateway in the system. The aim of this gateway is twofold:

- To provision the system with higher computation power. This allows consolidating in a single powerful computer the growing range of software functions that often present different safety and security implications and reducing in this way the overall number of control units present in the system. In addition, the increased performance allows to handle next generation of autonomous and intelligent systems that often rely on complex algorithms that demand high computation capabilities, as well as the execution of mixed-criticality functions.
- To enable the remote update of existing control devices in a secure way. The end-devices are commonly resource constrained (legacy) devices and therefore provide low computation capabilities. In this context, these devices might not be able to execute and enforce the required technical security functions. As compensating measure, these devices are deployed behind a security-aware gateway that manages their remote software updates enforcing defence-in-depth as required by IEC 62443. The gateway has the capability and flexibility to connect and update multiple and diverse end-devices, a solution that is scalable across different existing processors.

Therefore, the UP2DATE architecture is comprised of a high-performance gateway that connects to a server and multiple end-devices as depicted in Figure 2 that shows the update cycle explained below.

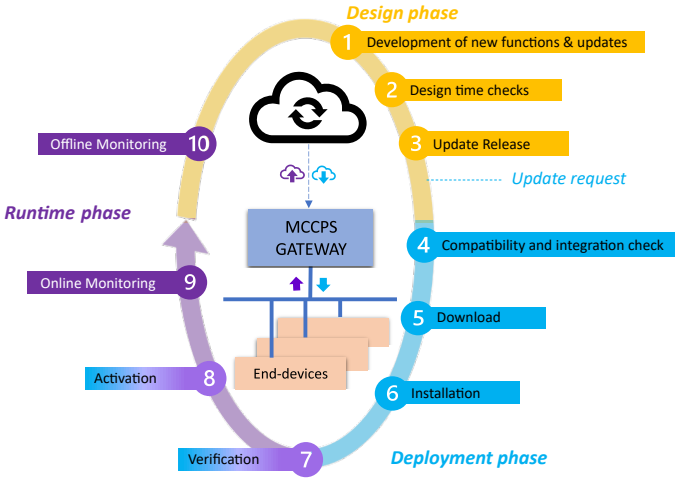


Fig. 2: Update cycle for mixed-criticality systems [9]

The update-cycle is comprised of 10 steps classified into three main phases [9]: (i) Design and release of updates, (ii) Update deployment phase and (iii) Runtime phase. The cycle starts at the moment at which the new or updated software component is available. Software modularity is adopted as the design principle to facilitate software modification in line with the recommendations of functional safety standards. A precondition is that each critical component is designed and developed according to the safety-security requirements for their target safety integrity and security level. In step 2, the design time checks are performed. After that, the software update is released (step 3). In the update deployment phase, compatibility and integration tests are carried out before authorizing update installation in the device. The update is then transferred from the server to the gateway and the installation accomplished. Just after, the correctness of the new software installation, configuration and dependencies with other software updates is verified.

TABLE I: MAGERIT [14] threat catalogue: industrial origin (**I**), failures and errors (**E**) and wilful attacks (**A**)

ID	Industrial origin (I)	ID	Failures and errors (E)	ID	Wilful attacks (A)
[I.1]	Fire	[E.1]	User's errors		
[I.2]	Water damage	[E.2]	Administrator errors		
[I.*]	Industrial disasters				
[I.3]	Mechanical pollution	[E.3]	Monitoring (logging) errors		
[I.4]	Electromagnetic pollution	[E.4]	Configuration error	[A.4]	Manipulation of the configuration
[I.5]	Hardware of software failure			[A.5]	Masquerading of user identity
[I.6]	Power interruption			[A.6]	Abuse of access privileges
[I.7]	Unsuitable temperature and/or humidity conditions	[E.7]	Organisation deficiencies	[A.7]	Misuse
[I.8]	Communications failure service	[E.8]	Malware diffusion	[A.8]	Malware diffusion
[I.9]	Interruption of other services and essential supplies	[E.9]	[Re-]routing errors	[A.9]	[Re-]routing of messages
[I.10]	Media degradation	[E.10]	Sequence errors	[A.10]	Sequence alteration
[I.11]	Electromagnetic radiation			[A.11]	Unauthorised access
				[A.12]	Traffic analysis
				[A.13]	Repudiation
		[E.14]	Information leaks	[A.14]	Eavesdropping
		[E.15]	Information alteration	[A.16]	Alteration of information
		[E.16]	Entry of incorrect information	[A.16]	Entry of false information
		[E.17]	Information degradation	[A.17]	Corruption of information
		[E.18]	Destruction of information	[A.18]	Destruction of information
		[E.20]	Software vulnerabilities		
		[E.21]	Defects in software maintenance/updating		
		[E.24]	System failure due to exhaustion of resources	[A.24]	Denial of service
				[A.25]	Theft
				[A.26]	Destructive attack
				[A.27]	Enemy over-run
		[E.28]	Staff shortage	[A.28]	Staff shortage
				[A.29]	Extorsion
				[A.30]	Social engineering

Finally, in the runtime phase, offline and online monitoring services are executed. On the one hand, online monitoring checks that the system meets its specification, and that safety and security metrics are within their safe and secure range at system operation. On the other hand, the offline monitoring service continuously sends data to a remote server for further analysis that serves to detect system malfunction.

IV. Context Establishment

Prior to the safety and security risk assessment, the context shall be defined, which includes the description for the system or product under test, as well as the circumstances and conditions in which the study is performed. This risk assessment focuses on a gateway and an end-device in the scope of an automotive use case. The system under evaluation, composed by these components, will provide functions such as diagnosis and safety, energy and thermal management, and driver interface among others. In addition, the gateway hosts diverse automotive grade domains such as Advanced Driver Assistance Systems (ADAS), In-vehicle infotainment (IVI), safety co-pilot etc. that are generally compute intensive and therefore need higher performance than that provided by regular automotive safety Electronic Control Units (ECUs).

The end-devices under consideration include safety functions and are therefore compliant with the ISO 26262 standard for Road Vehicle Functional Safety requirements, with the highest Automotive Safety Integrity Level (i.e., ASIL D). The gateway instead, can host both safety and non-safety related functions, following the previously defined mixed-criticality architecture on top of a certified hypervisor, that provides the required separation. In addition, this gateway also includes the update and monitoring middleware for update execution. It should be noted that in the scope of this analysis, the complete automotive case study is considered fail safe, i.e., a safe state can be reached either by the safety functions or diagnostics.

Concerning security, the system does not provide any security capability, except that a Virtual Private Network (VPN) is used for the communication of external entities with the gateway. Besides, the gateway and the end-devices are connected and communicated by a CAN bus. For the analysis, a single end-device is considered. Figure 3 shows a simplified application and deployment of the system. As depicted, an OBD-II connector, providing access to the internal bus is usually installed in the vehicle.

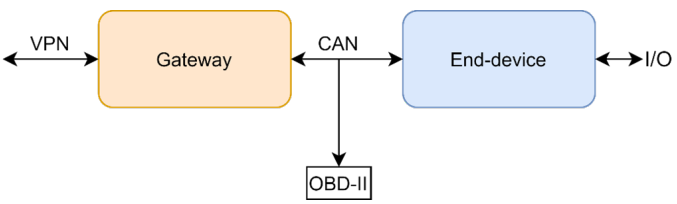


Fig. 3: Simplified application and deployment.

V. Safety & Security Risk Assessment

A risk assessment process is a systematic identification and evaluation process of all risks associated to a given scenario or purpose. The risk assessment processes play an important role in the safety and security management processes, since the identification and qualification of the safety failures, security threats and risks are essential when it comes to the protection of assets and people. This task shall be jointly addressed by all entities involved in the safety and security management process of the system under consideration.

A. Risk Identification

Risk identification is the process of determining the assets, the dependencies among them and the identification of safety and security threats associated to them (see threat catalogue in Table I), which may impact or compromise a given system property, denoted dimension. These dimensions are, according to MAGERIT [14]: *Confidentiality (C)*, *Integrity (I)*, *Availability (A)*, *Authenticity of service users (A_S)*, *Authenticity of data origin (A_D)*, *Accountability of service use (T_S)* and *Accountability of data access (T_D)*. Figure 4 depicts the adopted threat model.

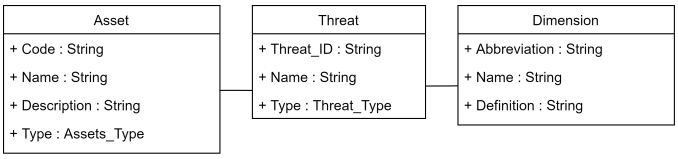


Fig. 4: Threat model (associations given by MAGERIT [14])

The identification and definition of the assets is an essential task to be performed in the risk assessment process. Assets are the resources included in the (sub)system or related to it that are necessary for the organisation (asset owner) to operate correctly and achieve the objectives proposed by its management. In this task, the asset classification and definitions provided by the MAGERIT [14] methodology are used. More particularly, the five type of assets described in Table II are considered.

It must be pointed out that these assets interplay jointly for achieving the goals of each use case. For instance, the risk analysis of services may depend on the analysis of other assets. Therefore, in this risk assessment process, the dependencies among assets in each use case are also examined. Following this asset categorization, on the one hand, Table III shows

TABLE II: Considered asset types [14]

Type	Abbrv.	Description
Services	[S]	"A function that meets a need of the users (of the service)."
Data / information	[D]	"Items of information which either individually or grouped together in some way, represent knowledge of something."
Software	[SW]	"With multiple names (programs, applications, developments, etc.) this section refers to tasks that have been automated and are carried out on a computer. Applications manage, analyze and change data, allowing the information to be used for providing services."
Computer equipment	[HW]	"Said to be material, physical goods, designed to directly or indirectly support the services provided and for the execution of computer applications".
Communication networks	[COM]	"Means of transporting data from one place to another."

TABLE III: Gateway asset list

Type	ID	Name	Description
[S]	G.S.01	Safety functions	Critical functions running on gateway partitions and that are required to avoid hazardous situations.
	G.S.02	Non-safety function	Any user application running on gateway partitions without safety implications.
	G.S.03	Software updating service	Service to update a software partition in the gateway
	G.S.04	Monitoring service	Service to monitor the gateway and its partitions.
[D]	G.D.01	Software update	Software update for the gateway.
	G.D.02	SASE properties	SASE properties of an update (either gateway or end-device).
	G.D.03	Monitoring data	Monitoring data of the gateway and its partitions.
[SW]	G.SW.01	Critical user software	Safety critical partitions running user applications on top of the hypervisor. Can be either bare-metal or include an Operating System (OS).
	G.SW.02	Non-critical user software	Non-critical partitions running user applications on top of the hypervisor. Can be either bare-metal or include an Operating System (OS).
	G.SW.03	Update middleware	Software responsible of managing the update of the gateway or external end-devices.
	G.SW.04	Monitoring middleware	Software responsible of monitoring the gateway or external end-devices.
	G.SW.05	Hypervisor	Software virtualization layer that provides separation on a mixed-criticality platform.
[HW]	G.HW.01	HPEC platform	High performance heterogeneous computing platform comprised of multicore CPUs and other types of accelerators such as GPUs.
[COM]	G.COM.01	VPN	Communication between server and gateway.

the identified assets in the gateway component. On the other hand, Table IV shows the identified assets in the end-device component.

TABLE IV: End-device asset list

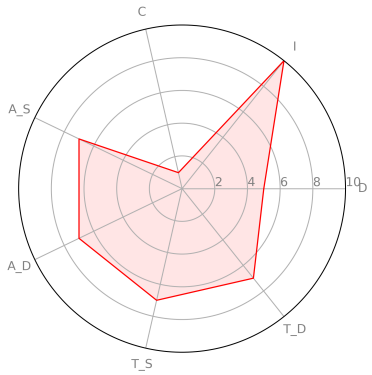
Type	ID	Name	Description
[S]	ED.S.01	Safety functions	Critical functions running on end-devices and that are required to avoid hazardous situations.
	ED.S.02	Software updating service	Service to update end-device software.
	ED.S.03	Monitoring service	Service to monitor the behaviour of the end-device.
[D]	ED.D.01	Software update	Software update for end-devices.
	ED.D.02	Monitoring data	Monitoring data of the end-device for update validation.
[SW]	G.SW.01	End-device software	All software running on the end-device, including the firmware, OS, bootloader, monitors and user applications.
[HW]	G.HW.01	End-device platform	Embedded computing platform.
[[COM]]	G.COM.01	CAN	Communication between gateway and end-devices or among multiple end-devices.

B. Risk Analysis

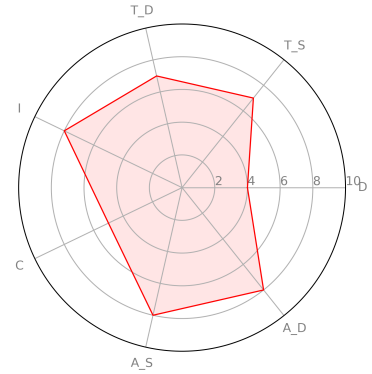
After the identification of assets, the security threats associated to such elements (shown in Table I) are determined. The association of safety and security threats to the assets, as well as impacted dimension, is accomplished depending on the asset type, as specified by MAGERIT [14]. In the risk analysis stage, the potential impact in each of the affected dimension and the failure/attack likelihood is estimated. To this end, two complementary strategies are used for safety and security. For safety, a simplified Failure Mode Effects and Critically Analysis (FMECA) is developed with the focus on the identified assets and the potential causes obtained from the MAGERIT catalogue. A Failure Mode and Effects Analysis (FMEA) is a systematic procedure for the analysis of a system in order to identify the potential failure modes, their causes and effects on system performance. FMECA is an extension to the FMEA to include means of ranking the severity of the failures modes to allow prioritization of countermeasures. This is done by combining the estimation of the severity of failure effects with a ranking estimation of the probability of the failure cause and the ability to detect potential failures on time.

On the contrary, in security, the computation of a fine-grained attack probability cannot be computed. Therefore, an attack likelihood estimation is done, in which an approximated attack potential is valued. To this end, several attack factors, such as required expertise, equipment, and window of exposure are considered.

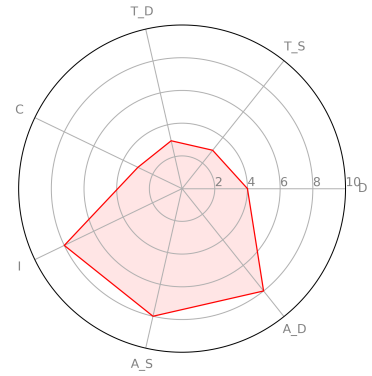
1) Potential Impact: The assessment of the potential impact is performed considering all the dimensions which might affect or compromise an asset. This metric is evaluated from 0 to 10, an impact of 0 implying no impact at all, while 10 indicating catastrophic consequences. Figure 5 shows the potential impact of *G.S.01* (5a), *G.SW.03* (5b) and *ED.COM.01* (5c) assets in all dimensions.



(a) Safety functions (G.S.01)

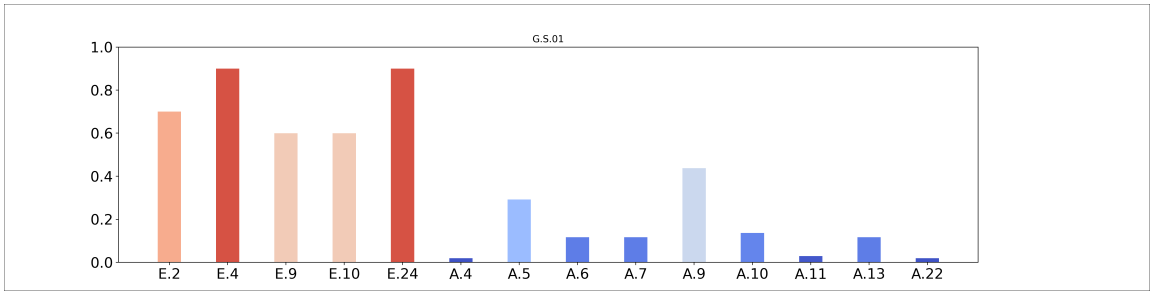


(b) Update middleware (G.SW.03)

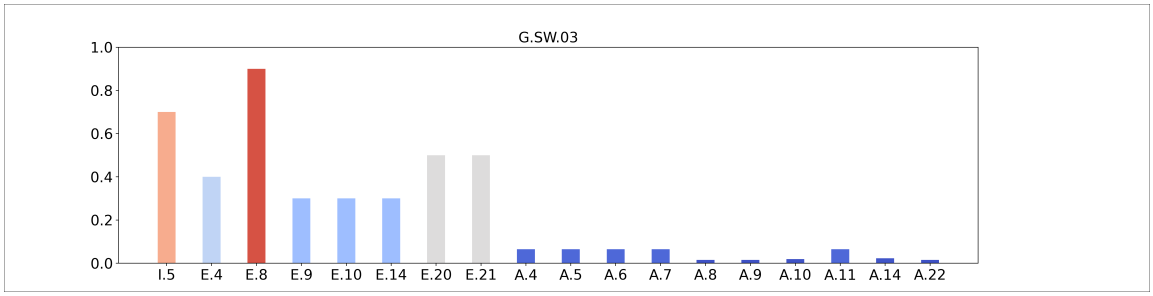


(c) CAN (ED.COM.01)

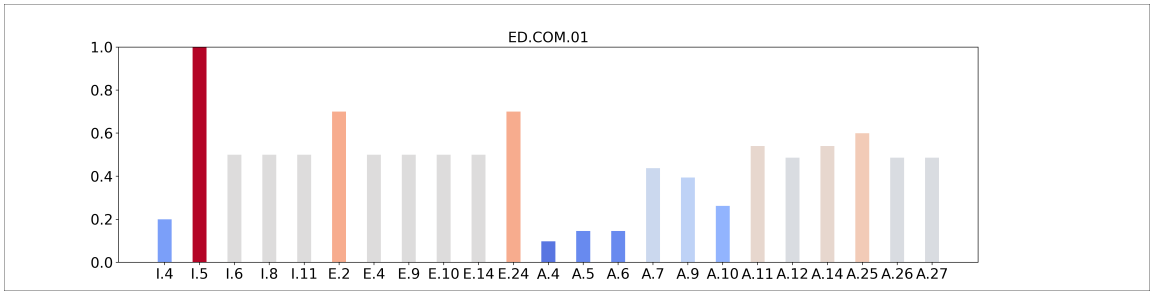
Fig. 5: Examples of potential impacts.



(a) Safety functions



(b) Update middleware



(c) CAN

Fig. 6: Examples of likelihood assessments.

The most impacted dimension is integrity, since an error or manipulation in the services, software and hardware can directly result in a critical system safety issue. Availability has, in general, a medium impact. While monitoring data and safety and security properties are crucial for safety function diagnostics, as the system is fail-safe, it could be moved to a safe state whenever this data is not available. Moreover, the authenticity of users and data origin is critical, any unauthorized access and use might seriously compromise the overall system safety. Finally, regarding, accountability, any undetected and unattended service use can cause major disruptions. Sometimes, it may also imply contract violations.

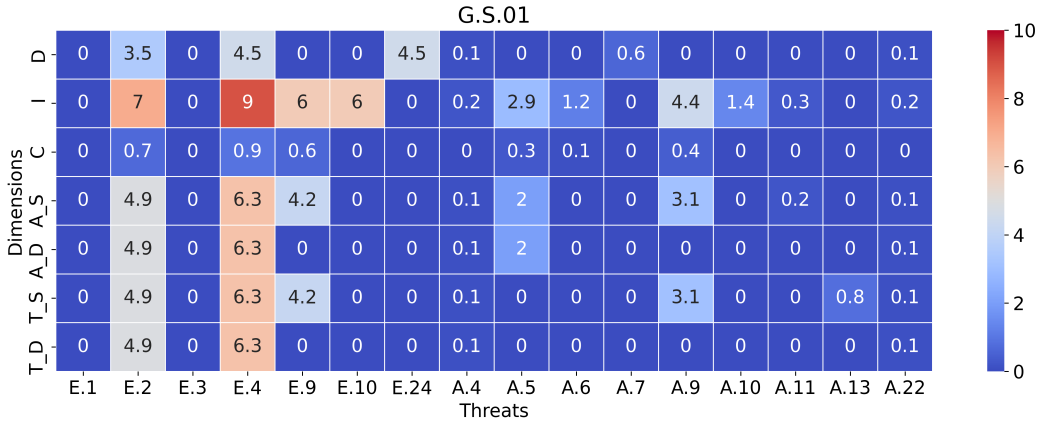
2) Likelihood: The assessment of likelihood is the estimation of the probability of failure or the effort required by an attacker to perform an attack. For this purpose, it is assumed that design and deployment best practices are applied, for example, disabling other communication protocols and closing unnecessary communication ports. For safety related failures, these values are obtained from a FMECA. The assumptions and the scenario described previously are also considered.

Figure 6 shows the failure and attack likelihood of *G.S.01* (6a), *G.SW.03* (6b) and *ED.COM.01* (6c) assets. As observed, unintentional failures and errors present, generally speaking, high likelihood, specially for E.2 (administrator errors), E.4 (configuration error), E.8 (malware diffusion) and E.24 (system failure due to exhaustion of resources). From the security point of view, the CAN bus (*ED.COM.01*) entails remarkable attack likelihood levels, notably for A.11 (unauthorised access), A.14 (eavesdropping) and A.25 (theft). Finally, the system is also highly susceptible to hardware and software failures (I.5).

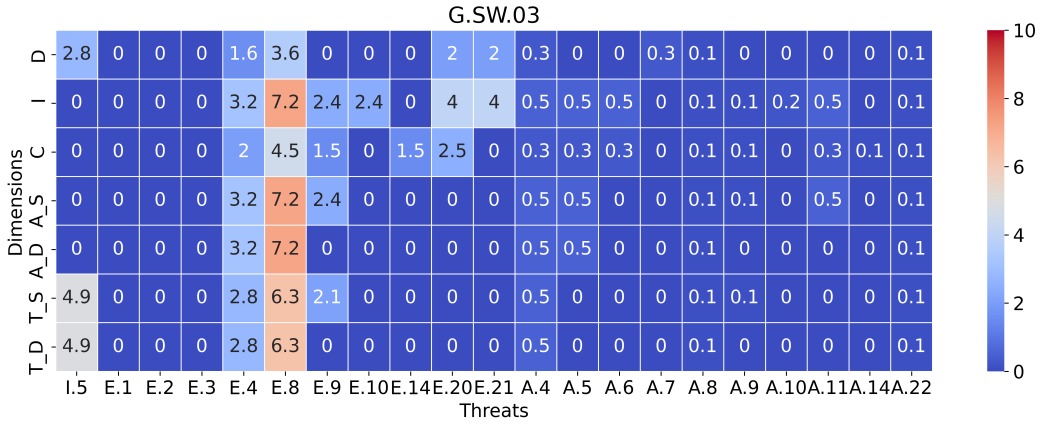
C. Risk Evaluation

In this step, the resulting risk associated to each asset is calculated. For it, the potential impact and failure/attack likelihood evaluation results are used. Risk is evaluated as shown in Equation 1 and measured from 0 to 10.

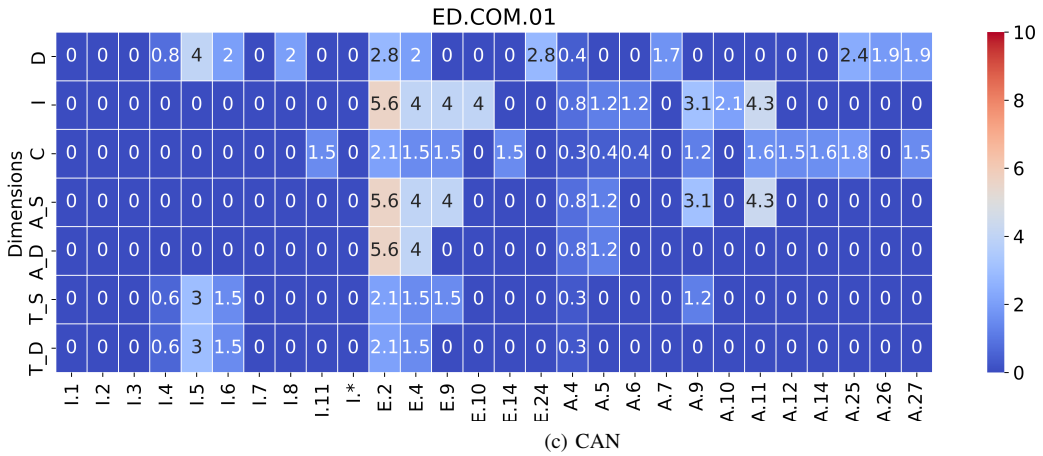
$$Risk_{A,D,T} = Potential\_Impact_{A,D,T} * Likelihood_{A,D,T} \quad (1)$$



(a) Safety functions



(b) Update middleware



(c) CAN

Fig. 7: Examples of evaluated risk maps.

A risk map for each asset is then built, which indicates the risk level in each dimension for each threat. The computed risk maps for the previously presented assets are shown in Figure 7. As can be seen, the gateway assets *G.S.01* (7a) and *G.SW.03* (7b) present high levels or risks for errors and failures. The authenticity, traceability and integrity properties of the component might severely be compromised. [Re-]routing of messages (A.9) attacks also represent a danger for *G.S.01*.

On the contrary, *ED.COM.01* (7c) presents medium risk levels which might jeopardize the integrity and authenticity dimensions. The security threat to be tackled is the unauthorised access (A.11) to the system. The CAN bus is also susceptible to administrator errors (E.2), configuration errors (E.4), [Re-]routing errors (E.9) and sequence errors (E.10).

VI. Risk Treatment

The goal of the risk treatment phase is to address the previously identified risks. Usually, safety and security measures are implemented to decrease them. Nevertheless, other strategies might be adopted if the cost of implementation is high, finding the appropriate balance. Therefore, through a risk versus cost analysis, a risk threshold is defined. All risks above such threshold shall be addressed and mitigated, while the risks below can be disregarded. At this stage, we consider three levels of risk: low (1 to 4), medium (5 to 7) and high (8 to 10). In this safety and security concept, failure detection or avoidance countermeasures, in addition to security measures are defined for all those threats entailing medium or high risk.

A. Functional Safety & Cyber-security Management

In order to avoid systematic faults during the different phases of the development process and to develop and maintain a secure product, the definition and enforcement of a Functional Safety and Cyber-security Management process is recommended. To this end, the safety and security methodology requirements from IEC 61508-1 (clause 6) [7] and IEC 62443-4-1 [10] should be considered.

B. Diagnostic Mechanisms

Runtime error detection is implemented through diagnostic mechanisms that achieve the required diagnostic coverage (DC) for each integrity level and architecture design. The particular measures could be selected from IEC 61508-2 and -3 Annex A according to the required diagnostic coverage [7]. At a high-level, the considered diagnostic mechanisms can be classified as follows:

- Autonomous hardware diagnostics: the hardware platform includes autonomous diagnostic mechanisms.
- Software-commanded diagnostics: the system includes hardware diagnostic components to be commanded by software including features for the diagnostic of independence violations.
- Platform independent diagnostics: additional diagnostics for hardware components and software applications.
- External diagnostics: system diagnostics external to the gateway, e.g., off-chip redundancy with majority voter or external watchdog for temporal and logical monitoring with independent clock source.
- Independence violations detection: Measures for the detection of independence violations are implemented by hardware diagnostic mechanisms (e.g., MPU, watchdog) and by the online monitoring that supervises the correct temporal behaviour of each partition and handles the external watchdog in case of execution time exceeding events.

C. Independence of Execution

Independence of execution is a crucial property of mixed-criticality systems, and it shall be guaranteed both on the spatial and temporal domains. To this end, based on previous work done in EU projects of the mixed-criticality cluster such as MultiPARTES [15], PROXIMA [16], DREAMS [17] and SAFEPOWER [18], the following services and mechanisms should be provided by the hypervisor: resource management, time synchronization, inter-partition communications, fault management and logging and safe system start-up and shutdown.

D. Safe and Secure Update

The update shall be deployed following a predefined safe and secure procedure. In order to guarantee safety and security, an important aspect of this procedure is the verification and validation of the changed software.

E. Safe and Secure Configuration

System configuration shall be defined at design time by safety and security system architects and programmed using qualified tools. The UP2DATE architecture has the particularity that the configuration may need to be adapted with a software update. However, in all cases, this configuration shall be defined and validated at design phase, and it shall be protected against unintended runtime modifications out of the updating process.

F. Compatibility and Integration Check

Compatibility and integration check is a crucial technique for the verification of new updates. The overall goal of this check is to verify that all software components meet with the constraints defined in their safety and security properties, which include requirements for their integration with existing software modules and with the hardware platform and its configuration.

G. Safe and Secure Communications

Different security zones will be connected through conduits that provide the security functions that enable the secure communication. All zone boundaries are supervised and managed through firewalls, in which a security policy is enforced. In these security policies (in each firewall) all network traffic shall be denied by default, and legitimate and required communications allowed.

H. Online Monitoring

Online monitoring will verify, based on runtime information, that the system meets its specification (and more specifically, its safety and security properties) before, during and after an update and that it is therefore operating within safe bounds according to the constraints of each compliant item. In this way, it is possible to detect residual specification and implementation faults in software and system integration faults.

This monitoring has a direct impact on system safety and therefore, the system shall be capable of reacting within the Process Safety Time (PST), that is, before a hazardous event is caused. This is the reason why online monitoring runs on the gateway itself (to mitigate the communication overhead of sending the data to an external device).

I. Offline Monitoring

Offline monitoring is used for security fingerprinting, which is devoted to the detection of performance anomalies that could result for instance, from malicious code installation during an update. It should be confirmed that the software update does not contradict the system reliability and operability features. For this purpose, the system is monitored in two phases, following the approach presented by *Cherkasova, Ludmila, et al.* [19].

J. Access Control Scheme

The access control scheme manages which entity, such a person or machine, is allowed to communicate and access the resources included in the system. For this purpose, these entities shall firstly be authenticated. On the one hand, the server and the gateway will use a Public Key Infrastructure from which the required authentication certificates will be generated. On the other hand, a symmetric cryptography-based challenge-response authentication mechanism is used for the authentication between the server and the end-device. Thus, a multi-factor authentication is required to perform an update in the end-device, a valid certificate for the connection to the gateway and a master key.

Besides, all inbound and outbound communications will be regulated by an integrated firewall in the gateway. All allowed secure communications and ports will be included in a whitelist. By default, all other communications will be blocked.

K. Security Auditor

The security auditor is a SCAP-enabled agent integrated within the system that is used to identify software flaws, vulnerabilities, and security-related misconfigurations. The Security Content Automation Protocol (SCAP) [20] is a group of standards defined by the National Institute of Standards and Technology (NIST) that enable the automated vulnerability management, measurement, and policy compliance evaluations. The auditor scans and checks (periodically or upon request) the system for vulnerabilities and weaknesses according to the SCAP specifications.

VII. Related work

Kavallieratos, G, *Sokratis K*, and *Vasileios, G* [21] presented a comprehensive survey of safety and cyber-security co-engineering methods. In this work, 25 methods related to safety and security risk analysis methods were analyzed. Nevertheless, as stated by the authors, a generic application and domain independent methodology should be used, such as the one defined by the ISO 27005 standard [12]. In this sense, four different well-known detailed risk analysis methods were studied by *Syalim, A*, *Yoshiaki H*, and *Kouichi S* [22], which are: Mehari, Magerit, NIST800-30 and Microsoft's Security Management Guide. Currently, Microsoft uses STRIDE methodology for products threat modelling [23].

The STRIDE methodology is widely used for the assets and threats identification. An IEC 62443 compliant risk analysis was presented by *M.Fockel et al.* [24] for the development of industrial control systems. This methodology was also used by *Zhendong Ma and Christoph Schmittner* for the threat modelling of connected and intelligent vehicles [25] and by *A. Vasenev et al.* [3] for a automotive case considering specific OTA threats. Nevertheless, it has to be pointed out that the STRIDE methodology addresses the system elements (assets) and threats identification, it does not cover the impact and likelihood estimations, nor the risk computation.

In order to tackle this gap, *J.P. Monteuuis et al.* from PSA Group, Telecom ParisTech and CEA LIST, propose the SARA framework for threat modelling and risk assessment for driverless vehicles. Although also based on STRIDE, the authors extend such method to define systematic threat analysis and risk assessment process, in which the safety issues are also considered. For this purpose, the severity, attack likelihood and controllability parameters are evaluated. SARA is divided into four main phases: (1) Feature definition; (2) Threat specification; (3) Risk assessment; (4) Countermeasures.

As far as integrated safety and security risk assessments are concerned, ABB [26] proposes a safety and security addressing methodology for safety-critical systems. As stated, safety and security should jointly be managed. This approach was also supported by *S. Plósz et al.* [27]. For the combined assessment, a combined catalogue composed by a failures catalogue (based on FMEA) and an attacks catalogue (based on STRIDE) was created. This method enables efforts saving, raising issues which may not be identified instead and multi-dimensional decision making. Finally, an argumentation case for safe and secure automotive OTA updates was presented by *T.Chowdhury et al.* [28].

VIII. Conclusions

Software-intensive safety-critical systems are facing new needs. Similar to consumer products, OTA updates could provide higher flexibility and maintainability capabilities, including security weaknesses and bugs fixing. However, it presents several technical challenges, as well as safety and security risks. Although required for security, software modifications and upgrades on safety-critical systems are commonly not

recommended. A safety re-certification may also involve high efforts and costs.

In this paper, a safety and security concept for software updates on mixed-criticality systems is presented. For this purpose, a safety and security risk assessment of a next generation automotive system is performed, composed by a gateway and an end-device. The safety and security analysis and measures defined in this concept will be further developed in the UP2DATE European project and validated at the automotive and railway case studies.

ACKNOWLEDGMENT

This work has partially funded by the European Union's Horizon 2020 research and innovation programme (grant agreement No 871465 (UP2DATE)).

REFERENCES

[1] I. Agirre, J. Abella, M. Azkarate-Askasua, and F. J. Cazorla, "On the tailoring of cast-32a certification guidance to real cots multicore architectures," in *2017 12th IEEE International Symposium on Industrial Embedded Systems (SIES)*. IEEE, 2017, pp. 1–8.

[2] C. Ebert and J. Favaro, "Automotive software," *IEEE Software*, vol. 34, no. 03, pp. 33–39, 2017.

[3] A. Vasenev, F. Stahl, H. Hamazaryan, Z. Ma, L. Shan, J. Kemmerich, and C. Loiseaux, "Practical security and privacy threat analysis in the automotive domain: Long term support scenario for over-the-air updates." in *VEHITS*, 2019, pp. 550–555.

[4] I. Mugarza, J. Parra, and E. Jacob, "Software updates in safety and security co-engineering," in *International Conference on Computer Safety, Reliability, and Security*. Springer, 2017, pp. 199–210.

[5] ——, "Cetratus: Towards a live patching supported runtime for mixed-criticality safe and secure systems," in *2018 IEEE 13th International Symposium on Industrial Embedded Systems (SIES)*. IEEE, 2018, pp. 1–8.

[6] I. Mugarza, J. L. Flores, and J. L. Montero, "Security issues and software updates management in the industrial internet of things (iiot) era," *Sensors*, vol. 20, no. 24, p. 7160, 2020.

[7] International Electrotechnical Commission and others, "Functional safety of electrical/electronic/programmable electronic safety related systems," *IEC 61508*, 2000.

[8] I. Agirre, P. Onaindia, T. Poggi, I. Yarza, F. J. Cazorla, L. Kosmidis, K. Grüttner, M. Abuteir, J. Loewe, J. M. Orbegozo *et al.*, "Up2date: Safe and secure over-the-air software updates on high-performance mixed-criticality systems," in *2020 23rd Euromicro Conference on Digital System Design (DSD)*. IEEE, 2020, pp. 344–351.

[9] I. Agirre, I. Yarza, I. Mugarza, J. Binchi, P. Onaindia, T. Poggi, F. J. Cazorla, L. Kosmidis, K. Gruttner, P. Uven, M. Abuteir, J. Loewe, J. M. Orbegozo, and S. Botta, "Management of automotive software updates," *Microprocessors and Microsystems*.

[10] International Electrotechnical Commission and others, "Iec 62443: Industrial communication networks - network and system security," *ed. GENEVA, SWITZERLAND: IEC Central Office*, 2010.

[11] "ISO 31000: Risk management — Guidelines," International Organization for Standardization, Geneva, CH, Standard, 2018.

[12] "ISO 27005: Information technology — Security techniques — Information security risk management," International Organization for Standardization, Geneva, CH, Standard, 2014.

[13] "Road vehicles — Cybersecurity engineering," International Organization for Standardization, Geneva, CH, Standard, 2021.

[14] Ministerio de Administraciones Públicas, Ed., *MAGERIT V.2: Analysis and Risk management information systems.* Ministry of finance and Public Administrations, 2006. [Online]. Available: https://administracionelectronica.gob.es/

[15] S. Trujillo, A. Crespo, A. Alonso, and J. Pérez, "Multipartes: Multicore partitioning and virtualization for easing the certification of mixed-criticality systems," *Microprocessors and Microsystems*, vol. 38, no. 8, pp. 921–932, 2014.

[16] R. Davis, T. Vardanega, J. Alexanderson, V. Francis, P. Mark, B. Ian, A.-A. Mikel, F. Wartel, L. Cucu-Grosjean, P. Mathieu *et al.*, "Proxima: a probabilistic approach to the timing behaviour of mixed-criticality systems," *Ada User Journal*, vol. 2, no. 118-122, p. 181, 2014.

[17] A. Larrucea, I. Martinez, V. Brocal, S. Peirò, H. Ahmadian, J. Perez, and R. Obermaisser, "Dreams: Cross-domain mixed-criticality patterns," in *Workshop on Mixed-Criticality Systems*, 2016, p. 6.

[18] M. Fakih, A. Lenz, M. Azkarate-Askasua, J. Coronel, A. Crespo, S. Davidmann, J. C. D. Garcia, N. G. Romero, K. Grüttner, S. Schreiner *et al.*, "Safepower project: Architecture for safe and power-efficient mixed-criticality systems," *Microprocessors and Microsystems*, vol. 52, pp. 89–105, 2017.

[19] L. Cherkasova, K. Ozonat, N. Mi, J. Symons, and E. Smirni, "Automated anomaly detection and performance modeling of enterprise applications," *ACM Transactions on Computer Systems (TOCS)*, vol. 27, no. 3, pp. 1–32, 2009.

[20] D. Waltermire, S. Quinn, K. Scarfone, and A. Halbardier, "The technical specification for the security content automation protocol (scap): Scap version 1.2," *NIST Special Publication*, vol. 800, p. 126, 2011.

[21] G. Kavallieratos, S. Katsikas, and V. Gkioulos, "Cybersecurity and safety co-engineering of cyberphysical systems—a comprehensive survey," *Future Internet*, vol. 12, no. 4, p. 65, 2020.

[22] A. Syalim, Y. Hori, and K. Sakurai, "Comparison of risk analysis methods: Mehari, magerit, nist800-30 and microsoft's security management guide," in *2009 International conference on availability, reliability and security*. IEEE, 2009, pp. 726–731.

[23] A. Shostack, "Experiences threat modeling at microsoft." *MODSEC@ MoDELS*, vol. 2008, 2008.

[24] M. Fockel, S. Merschjohann, M. Fazal-Baqaie, T. Förder, S. Hausmann, and B. Waldeck, "Designing and integrating iec 62443 compliant threat analysis," in *European Conference on Software Process Improvement*. Springer, 2019, pp. 57–69.

[25] Z. Ma and C. Schmittner, "Threat modeling for automotive security analysis," *Advanced Science and Technology Letters*, vol. 139, pp. 333–339, 2016.

[26] F. Reichenbach, J. Endresen, M. M. Chowdhury, and J. Rossebø, "A pragmatic approach on combined safety and security risk analysis," in *2012 IEEE 23rd International Symposium on Software Reliability Engineering Workshops*. IEEE, 2012, pp. 239–244.

[27] S. Plósz, C. Schmittner, and P. Varga, "Combining safety and security analysis for industrial collaborative automation systems," in *International Conference on Computer Safety, Reliability, and Security*. Springer, 2017, pp. 187–198.

[28] T. Chowdhury, E. Lesiuta, K. Rikley, C.-W. Lin, E. Kang, B. Kim, S. Shiraishi, M. Lawford, and A. Wassyng, "Safe and secure automotive over-the-air updates," in *International Conference on Computer Safety, Reliability, and Security*. Springer, 2018, pp. 172–187.