Deliverable 7.2

Security Certification Methodology Definition

Technical References

Document version	:	1.0
Submission Date	:	28/02/2022
Dissemination Level	:	Public
Contribution to	:	WP7 – Security and Privacy Claims
Document Owner	:	UMU
File Name	:	BIECO_D7.2_28.02.2022_V1.0
Revision	:	3.0

Project Acronym	:	BIECO
Project Title	:	Building Trust in Ecosystem and Ecosystem Components
Grant Agreement n.	:	952702
Call	:	H2020-SU-ICT-2018-2020
Project Duration	:	36 months, from 01/09/2020 to 31/08/2023
Website	:	https://www.bieco.org

This project has received funding from the European Union´s Horizon 2020 Research and Innovation Programme under Grand agreement No. 952702.

Revision History

REVISION	DATE	INVOLVED PARTNERS	DESCRIPTION
0.1	01/02/2021	UMU	First draft of the structure
0.2	06/04/2021	UMU	Inputs of SOTA
0.3	12/04/2021	CNR	Toc changes
0.4	3/5/2021	UMU	Challenges and review to comments from partners.
0.5	25/06/2021	UMU	New sections 3 and 4 added
0.6	20/09/2021	UMU	New content about SOTA and methodology
0.7	15/11/2021	RES	Contributions to SOTA
0.8	23/11/2021	UMU	Contributions to SOTA
0.9	25/11/2021	UMU	Refinement of risk identification and testing
1.0	28/11/2021	CNR	Draft of privacy certification. Contributions to SOTA on security testing (added Sections 4.6 and 4.7). Added content to Challenges of security certification and evaluation (Sec. 5.12).
1.1	20/12/2021	UMU	Contributions to Section 6
1.2	30/12/2021	UMU	Finish methodology definition, conclusions, summary, artefacts
1.3	10/01/2021	UMU	Corrected some typos
1.4	14/01/2022	7B	Internal review and fixing minor issues.
1.4	17/01/2022	IESE	Contribution to the State of the Art with Section 3.5 and its subsections on Safety & Security Risk Analysis Techniques
1.5	17/01/2022	GRAD	General review and fixing minor issues
1.6	19/01/2022	7B	Contributions to Section 6, risk calculation alternatives. Fixed minor issues in the text.
1.6	19/01/2022	UMU	Table and images references, bibliography, acronyms table, fixing minor issues. General review.
1.7	01/02/2022	UNINOVA	Internal Review
1.8	02/02/2022	UMU, IESE	Fixing review issues
1.9	09/02/2022	IFEVS	Internal Review
2.0	13/02/2022	UMU	Fixing review issues
2.1	17/02/2022	7Bulls, UMU	Final review of WP leader
2.2	25.02. 2022	UNI	Final Revision and correction by Coordinator
3.0	28.02.2022	UNI	Finalizing deliverable and submission

List of Contributors

Deliverable Creator(s): Sara N. Matheu (UMU), Eda Marchetti (CNR), Said Daoudagh (CNR), Radosław Piliszek (7B), Enrico Schiavone (RES), Eva Sotos (GRAD), Emilia Cioroaica (IESE), Adrián Sánchez (UMU).

Reviewer(s): Ana Inês Oliveira (UNI), Riccardo Introzzi (IFEVS); Sanaz Nikghadam-Hojjati (UNI); José Barata (UNI).

BIECO project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No **952702**.

Acronyms

Acronym	Term
AIAG	Automotive Industry Action Group
ANSI	American National Standards Institute
ANSSI	National Cybersecurity Agency of France
ASIL	Automotive Safety Integrity Level
C2C-CC	Car 2 Car Communication Consortium
CC	Common Criteria
CCRA	Common Criteria Recognition Arrangement
CESG	Communications-Electronics Security Group
CHASSIS	Combined Harm Assessment of Safety and Security for Information Systems
CPA	Commercial Product Assurance
cPP	Collaborative Protection Profiles
CSA	Cybersecurity Act
CSPN	Certification de Sécurité de Premier Niveau
CVE	Common Vulnerabilities and Exposures
CVSS	Common Vulnerability Scoring System
CWE	Common Weakness Enumeration
CWE	Common Weakness Enumeration
CWRAF	Common Weakness Risk Analysis Framework
CWSS	Common Weakness Scoring System
D-MUC	Diagrammatic Misuse Cases
DoS	Denial of Service
DPIA	Data Protection Impact Assessment
DREAD	Damage potential, Reproducibility, Exploitability, affected users, Discoverability
DSL	Domain Specific Languages
D-UCs	Diagrammatic Use Cases
EAL	Evaluation Assurance Levels
EC	European Commission
ECSC	European Cyber Security Certificate
ECSO	European Cyber Security Organisation
ENISA	European Union Agency for Network and Information Security
ETSI	European Telecommunications Standards Institute
EU	European Union
EUCC	Common Criteria based European candidate cybersecurity certification scheme
EUCS	European Cybersecurity Certification Scheme for Cloud Services
EVITA	E-safety Vehicle InTrusion protected Applications
FMEA	Failure Mode and Effect Analysis
FMVEA	Failure Mode, Vulnerabilities and Effects Analysis
FSD	Failure-Sequence Diagrams
GDPR	General Data Protection Regulation
HARA	Hazard Analysis and Risk Assessment
HARM	Hailstorm Application Risk Metric
HAZOP	Hazard and Operability studies
HEAVENS	HEAling Vulnerabilities to ENhance Software Security and Safety
IACS	Industrial Automation and Control Systems
ICT	Information and Communications Technology
IEC	International Electrotechnical Commission

ISA	International Society of Automation
ISO	International Organization for Standardization
ITU	International Telecommunication Union
MBST	Model-Based Security Testing
MBT	Model-Based Testing
MIA	Model Inference Algorithm
MRA	Mutual Recognition Agreement
MUD	Manufacturer Usage Description
MUSD	Misuse sequence Diagrams
NCCA	National Cybersecurity Certification Authority
NIST	National Institute of Standards and Technology
NVD	National Vulnerability Database
OCTAVE	Operationally Critical Threat, Asset, and Vulnerability Evaluation
OSSTMM	Open-Source Security Testing Methodology Manual
OWASP	Open Web Application Security
OWASP	Open Web Application Security Project
PP	Protection Profile
QR	Quick Response
RTU	Remote Terminal Units
SAHARA	Security-Aware Hazard Analysis and Risk Assessment
SANS	SysAdmin, Audit, Network and Security
SAST	Static Application Security Testing
SCA	Static Code Analysis
SCADA	Supervisory control and data acquisition
SDLA	Security Development Lifecycle Assurance
SecL	Security Level
SOG-IS	Senior Officials Group Information Systems Security
SOTA	State Of the Art
SSA	System Security Assurance
ST	Security Target
STAMP	System Theoretic Accident Model and Processes
STPA	System Theoretic process Analysis
STPA-Sec	Systems-Theoretic Process Analysis for Security
STRIDE	Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, elevation of Privileges
SUT	System Under Test
TAL	Trust Assurance Levels
T-MUC	Textual Misuse Cases
TOE	Target of Evaluation
TP	Tolerance Profiles
TTCN	Testing and Test Control Notation
T-UCs	Textual Use Cases
TVRA	Threat, Vulnerability, and Risk Analysis
UL	Underwriters Laboratories
UML	Unified Modelling Language
UNECE	United Nations Economic Commission for Europe

Executive Summary

This deliverable reports the work done in T7.2, whose purpose is the definition of a security evaluation methodology to evaluate the security of an Information and Communications Technology (ICT) system. The methodology is based on standards such as International Organization for Standardization (ISO) 31000 standard for Risk Management, the ISO 29119 standard for Security Testing or the Manufacturer Usage Description (MUD) standard. The methodology defines a set of high-level steps that should be followed by the security evaluator and is intended to serve as a basis for the security certification.

The deliverable analyses and identifies challenges associated with current certification schemas, providing a methodology that addresses several of the identified challenges. In particular, the combination of risk assessment and testing processes provides an objective and empirical measurement that also allows a partial automation of the process, facilitating subsequent recertification in case there is a security change in the system. The methodology also considers the context variability (different security level in different contexts), the definition of a visual label for non-expert consumers and the creation of a behavioural profile with security recommendations to address encountered issues.

This task has been closely related to T7.1, as the methodology uses as input the set of claims defined in D7.1 to evaluate the security of a system. Additional collaborations with other WPs have been also envisioned during the definition of the methodology, for example WP3 and WP6 for testing and risk assessment or WP5 for monitoring. Indeed, the behavioural profile created at the end of the process is intended to update the extended MUD file developed in WP6 and serve as input for the monitoring of suspicious behaviours in WP5. Further collaborations will be also analysed in T7.3. This task is going to serve as input for the implementation of the methodology in T7.3, which will provide a concrete example of how the methodology could be instantiated.

Project Summary

Nowadays most of the ICT solutions developed by companies require the integration or collaboration with other ICT components, which are typically developed by third parties. Even though this kind of procedures are key in order to maintain productivity and competitiveness, the fragmentation of the supply chain can pose a high-risk regarding security, as in most of the cases there is no way to verify if these other solutions have vulnerabilities or if they have been built taking into account the best security practices.

In order to deal with these issues, it is important that companies make a change on their mindset, assuming an "untrusted by default" position. According to a recent study only 29% of IT business know that their ecosystem partners are compliant and resilient with regard to security. However, cybersecurity attacks have a high economic impact, and it is not enough to rely only on trust. ICT components need to be able to provide verifiable guarantees regarding their security and privacy properties. It is also imperative to detect more accurately vulnerabilities from ICT components and understand how they can propagate over the supply chain and impact on ICT ecosystems. However, it is well known that most of the vulnerabilities can remain undetected for years, so it is necessary to provide advanced tools for guaranteeing resilience and also better mitigation strategies, as cybersecurity incidents will happen. Finally, it is necessary to expand the horizons of the current risk assessment and auditing processes, taking into account a much wider threat landscape. BIECO is a holistic framework that will provide these mechanisms in order to help companies to understand and manage the cybersecurity risks and threats they are subject to when they become part of the ICT supply chain. The framework, composed by a set of tools and methodologies, will address the challenges related to vulnerability management, resilience, and auditing of complex systems.

Partners


Disclaimer

Table of Contents

List of Figures

List of Tables

1. Introduction

Nowadays, there is an increasing interest to establish a general basis for European security certification and labelling led by the European Union Agency for Network and Information Security (ENISA) through the cybersecurity act (CSA). The term certification is described by the NIST [1] as "*a comprehensive assessment of the management, operational, and technical security controls in an information system, made in support of security accreditation, to determine the extent to which the controls are implemented correctly, operating as intended, and producing the desired outcome with respect to meeting the security requirements for the system*". As a result of this process, it is expected to obtain a cybersecurity label (labelling process) which contains "*information that represents or designates the value of one or more security relevant-attributes''*. Despite well-known initiatives from European organizations, such as DIGITALEUROPE [2] or ECSO [3], to provide some guidelines towards certification, a certification and evaluation methodology must overcome different obstacles. On the one hand, the high degree of diversity and heterogeneity of devices and products conflicts with the need for objective comparisons regarding security aspects. On the other hand, due to the security dynamism, the certification methodology must consider these changing conditions, managing the device lifecycle and taking into account the context in which the system will be operating. Indeed, the CSA emphasizes the need for security approaches addressing the lifecycle of any ICT product, service, or process for the definition of a cybersecurity certification framework. Therefore, agile self-assessment schemes and test automation environments should be created and evolved to ensure products have a minimum-security level appropriate for a context where they are used. This deliverable analyses the flaws and strengths of the main cybersecurity certification schemes to provide a list of challenges that are still pending to be addressed.

Some initiatives such as ETSI [5] or ARMOUR [6] already put some effort to address some of the mentioned challenges. In particular, these approaches are characterized by combining risk assessment and testing following a test-based security risk assessment, in which testing (ISO 29119) is used to guide and improve the security risk assessment (ISO 31000), adjusting risk values and providing feedback. However, the ETSI approach only gives some high-level guidelines and ARMOUR focuses on an isolated IoT device.

This deliverable aims to define a security evaluation methodology based on these two approaches, that is, combining risk assessment and testing, with the objective of coping with most of the analysed challenges. The result is intended to serve as a basis for security certification, providing a generic set of steps that could be implemented through different testing techniques and tools.

The document is organised as follows: Section 2 performs a GAP analysis of the main security certification schemes. As the approach is based on two main blocks, security risk assessment and testing, Section 3 provides an analysis of the security risk assessment mechanisms and Section 4 does the same with the security testing techniques, identifying strengths and weaknesses of each approach. As a result, Section 5 collects the main identified challenges related to the security evaluation and certification of a system. Section 6 describes the security evaluation methodology proposed to deal with these challenges, as well as the high-level steps that the security evaluator should take to evaluate the security of a system. Section 7 summarises the conclusions obtained from the deliverable, Section 8 presents an overview of the artefacts produced in the deliverable and Section 9 lists the references.

2. GAP Analysis of Current Security Certification Approaches

This section presents an overview of the current certification methodologies, paying special attention to those which are standardized.

2.1 Common Criteria

The most well-known cybersecurity certification standard is the Common Criteria (CC) [7], in which the security functional and assurance requirements are specified through Protection Profiles (PPs) for a Target of Evaluation (TOE), which is a set of software, firmware and/or hardware. These requirements are defined in the Security Target (ST) description.

CC provides a set of common requirements for the security functionality of IT products, facilitating the comparability between the results of independent cybersecurity evaluations applied to these IT products. This is obtained through the Collaborative Protection Profiles (cPP), which become available for use under the terms of the Common Criteria Recognition Arrangement (CCRA) [8]. Evaluations conducted against cPPs on this list are mutually recognized according to the terms of the CCRA. The evaluation process establishes a level of confidence that the security functionality of these IT products and the assurance measures applied to these IT products meet these requirements. The evaluation results may help consumers to determine whether these IT products fulfil their security needs. For this purpose, it uses Evaluation Assurance Levels (EAL) to describe numerically the depth and rigor of an evaluation. Basically, CC assurance is achieved by carrying out analysis and checking of processes and procedures, guidance documents, TOE design, functional tests, independent functional testing, vulnerabilities, and penetration testing [9].

Although CC is the main standard, the community has identified several limitations [10][11] that are being considered. Examples of them are the time and effort required to document the evaluation and to gather evidence, in particular at high EALs, or the management of changes in the certified product. During the manufacturing process, this could involve market delays and, therefore, considerable financial loss. It is worth noting that CC evaluation is focused on a specific version of the TOE, including the configurations. This means that any change over in the TOE (e.g. a new vulnerability) could invalidate the result of the certification, something that is quite critical in frequently updated products. In addition, recertification is not mandatory in CC, and the responsibility of informing about security changes lies with the owner of the certificate.

2.2 Cyber Security Certification: EUCC Candidate Scheme

The European Union agency for cyber security, ENISA[12], is working on the first European cyber security certification scheme, based on CC and called "Common Criteria based European candidate cybersecurity certification scheme" (EUCC Scheme). The EUCC scheme aims to replace the existing schemes operating under the Senior Officials Group Information Systems Security (SOG-IS) Mutual Recognition Agreement (MRA) to add new elements and extend the scope to all EU Member States.

The EUCC scheme includes guidelines for its transition from CC, which is expected to be during the year 2022. It covers the assessment of generic ICT products through the 'Substantial' and 'High' EALs. The certificate is valid for five years and renewable. This scheme not only has recognition in all EU Member States, but also allows composite certification [3], that is, certification combining different schemes. Moreover, it uses a framework-based label and QR code to ensure easy access to certification information which is one of the main novelties for the end user. The scheme also harmonises the conditions for the management and disclosure of vulnerabilities, clearly defining rules on the monitoring and management of compliance and non-compliance and introducing a new patch management mechanism to support vulnerability management.

Although this initiative is a the first of a series of steps that Europe is doing towards a unified and harmonised certification, currently, the document focuses mainly on the certification framework, whereas concrete evaluation details, specially related to security testing and rating, are slightly mentioned. Indeed, the EUCC document, explicitly says that “groups of experts involving National Cybersecurity Certification Authority (NCCA), CAB and their testing facilities, and manufactures or providers of ICT products should be considered as to further develop harmonised requirements for the scheme”. In particular, “the expert groups should focus on methodology harmonization of testing, analysis of new attacks and applicability to ICT products (rating, updating of test methodology), and propose new or revised supporting documents (…) in order to cover any generic and specific domain”.

It is worth noting that the EUCC is the first European scheme developed in the frame of the CSA, but another scheme is being developed for cloud services (European Cybersecurity Certification Scheme for Cloud Services, EUCS) [13], and there are future plans for the 5G [14] and IoT domain.

2.3 Certification de Sécurité de Premier Niveau: CSPN

The Certification de Sécurité de Premier Niveau (CSPN) [15] is a French standard created by the National Cybersecurity Agency of France (ANSSI) in 2008. CSPN ensures independence through the auditors, who must be accredited by the ANSSI. The objective of this scheme is to verify the compliance of the products in relation to their specifications, assessing against the known vulnerabilities and stressing the product with tests aiming to break its security. The evaluation includes conformity analysis (verifying that the product complies with its security specifications) and efficiency analysis (measuring the strength of the security functions and mechanisms). One of the key points of CSPN is that the evaluation is performed in a short period of time through the adaptation of the product development lifecycle, reconciling the time needs of the manufacturing with the security assessment.

The CSPN approach starts by defining the Security Target (ST), describing the scope of the evaluation. After that, the ST is validated by the ANSSI and all the needed material is gathered by the evaluator (e.g., source code, a functional version of the product, historical data, and documentation). Then, the product is evaluated, allowing exchanges between the evaluator and the developers by providing extra material or giving information. This step includes the redaction of a detailed report that should include the test considered during the certification process, their results, and the uncovered issues.

After the evaluation, the conclusions of the report are validated by the ANSSI. The final step is the delivery of the CSPN certificate by the ANSSI in case of a successful outcome of all the previous steps [16].

CSPN is complementary to CC and can be used as a previous short and non-expensive assessment to CC certification [17].

2.4 Underwriters Laboratories 2900

Underwriters Laboratories (UL) is a company that certifies that electrical, building, fire, mechanical and other products follow their UL standards. In 2016, the company created the UL 2900 standard process [18]. This series of standards aims to provide a series of technical criteria to evaluate the security of a TOE.

However, these standards were created by a for-profit enterprise, and as they were not published, the research community could not validate them. This raised a lot of criticism. Also, due to its recent creation, it is not widely recognized as an accepted certification scheme. The standard has three parts: Outline of Investigation for Software Cybersecurity for Network-Connectable Products, General Requirements (UL 2900-1), Outline of Investigation for Software Cybersecurity for Network-Connectable Products, Particular Requirements for Network Connectable Components of Healthcare Systems (UL 2900-2-1) and Outline of Investigation for Software Cybersecurity for Network-Connectable Products, Particular Requirements for Industrial Control Systems (UL2900-2-2) that apply to the evaluation of industrial control systems components, such as Process control systems, Control servers, Supervisory Control And Data Acquisition (SCADA) servers, Remote Terminal Units (RTU) or smart sensors, among others. However, only the first part has been published as an American National Standards Institute (ANSI) standard.

Regarding recertification, in case there is a major change, the product must be completely certified again. There is no lightweight alternative process.

2.5 Commercial Product Assurance

The Commercial Product Assurance (CPA) [19] is the UK national scheme from the Communications-Electronics Security Group (CESG) in charge of the assessment of the security level of a TOE, testing and certifying both software and hardware within the UK government [20]. CPA Security Characteristics against the products are published in [21].

If a product passes the CPA assessment, it is awarded with the Foundation Grade certificate, valid for two years, and allowing any type of update required, leading in some way with the dynamicity of the security changes. However, recertification will only happen if a security change is known due to an update and the manufacturer wants to recertificate it [20].

The main barrier of CPA is that there is no Mutual Recognition Agreement (MRA) for it, meaning that, if a product was certified in UK, it will not usually be accepted abroad. However, this is being addressed through the so-called CPA mapping for the Protection

Profile that was used in the product CC evaluation, which performs a mapping between the protection profile of CC and the security statements of CPA [22].

2.6 ANSI ISA/IEC 62443 Cyber-Security Certification Programs

The International Society of Automation (ISA) / International Electrotechnical Commission (IEC) 62443 standard series [23], developed by the ISA committee and adopted by the IEC, provides a flexible framework for addressing and mitigating current and future security vulnerabilities in industrial automation and control systems (IACS).

This standard is arranged in four groups (general concepts, policies and procedures, system and component), and each of them is composed of parts that refine different aspects (e.g. lifecycle, patch management, risk assessment, etc.). It provides the detailed security requirements for control of automation system components starting from seven general requirements defined as "Fundamental requirements" and described in IEC TS 62443-1. In particular, ISA / IEC 62443 includes a certification program, the ISA Secure Certification [24]. The *ISA/IEC-62443-2-2 – Security for Industrial Automation and Control Systems – Part 2-2: IACS security program ratings* specify a methodology for the evaluation of security for each Zone in an IACS Automation Solution. A taxonomy for the Security Program Rating is included, which is a combination of the Security Level of technical security measures.

ISA Security Compliance Institute is a non-profit organization that has developed several product certifications programs for Controls Systems and Components and manages the ISASecure conformance certification program. Currently available ISASecure certification programs are:

- Security Development Lifecycle Assurance (SDLA) which certifies that the Security Lifecycle of a Product Supplier meets the requirements in Part 4-1.
- System Security Assurance (SSA) which certifies that Control System products have the capability to meet the requirements in Part 3-3 and have been developed in accordance with an SDLA program.
- Component Security Assurance, which certifies that Component products have the capability to meet the requirements in Part 4-2 and have been developed in accordance with an SDLA program. Certified Component products can be: Embedded Devices, Host Devices, Network Devices, and Software Applications.

3. Security Risk Assessment Analysis

Risk assessment is defined in CNSSI-4009 [25] as "*the process of identifying, prioritizing, and estimating risks. This includes determining the extent to which adverse circumstances or events could impact an enterprise. Uses the results of threat and vulnerability assessments to identify risk to organizational operations and evaluates those risks in terms of likelihood of occurrence and impacts if they occur*". In this sense, this document considers a risk assessment methodology as a process to determine the risk of a vulnerability, weakness, or threat.

It is worth noting that, although D6.1 provides a SOTA of current risk assessment schemes, the next subsections analyse the weaknesses and strengths of the main risk assessment schemes from a certification usability point of view (e.g. considering the objectivity of the process, the facility to calculate the metrics employed, etc.). The main objective of this analysis is to gather a set of certification-and evaluation-related challenges and study their feasibility.

3.1 Common Weakness Scoring System (CWSS)

The Common Weakness Scoring System (CWSS) [26] provides a mechanism for prioritizing software weaknesses and assigning a numerical risk to them. To do so, CWSS combines three groups of metrics that are used to calculate the risk: Base Finding, Attack Surface, and Environmental. The Base Finding is focused on the inherent risk of the weakness, the confidence in the accuracy of the finding, and the strength of controls, the Attack Surface, which includes factors representing the barriers that an attacker must exceed to exploit the weakness and lastly, the Environmental, which groups characteristics of the weakness that are specific to a particular environment or operational context.

Each factor in a metric group is assigned a value, which is converted to its associate weight. The metrics of each group are calculated and combined with the other groups (multiplication) to obtain a complete risk measure, which ranges between 0 and 100. The Base Finding subscore is between 0 and 100, whereas the other ones can range between 0 and 1. The main advantage of CWSS is that it allows unknown values when the information is incomplete, so it can be applied earlier in the process, before any vulnerability has been proven, [27].

If the proposed set of values for the Technical Impact metric is not precise enough, CWSS users can use their own quantified methods to derive a subscore. One of the methods uses the Common Weakness Risk Analysis Framework (CWRAF) to define a vignette and a Technical Impact Scorecard. Here, vignette-specific Importance ratings are used to calculate the Impact weight. CWRAF and CWSS allow users to rank classes of weaknesses independent of any particular software package in order to prioritize them relative to each other (e.g., "buffer overflows are higher priority than memory leaks"). This approach, sometimes referred to as a "Top-N list," is used by the Common Weakness Enumeration (CWE) / (SysAdmin, Audit, Network and Security (SANS) Top 25 and Open Web Application Security Project (OWASP) Top Ten.

CWSS is recommended by the International Telecommunication Union (ITU), due to its accuracy to reflect the risk to the user of the software capability, given the unique business context it will function within for the user, and the unique business capability the software is providing to the user [28][29].

3.2 Common Vulnerability Scoring System (CVSS)

The Common Vulnerability Scoring System (CVSS) [30] is an open framework for communicating the characteristics and severity of software vulnerabilities, similar to CWSS, since it also consists of three metric groups: Base, Temporal, and Environmental. The Base group represents the intrinsic qualities of a vulnerability, the Temporal group reflects the characteristics of a vulnerability that changes over time, and the Environmental group represents the characteristics of a vulnerability that are unique to the user environment, leading with the context influence challenge. The Base metrics produce a score ranging from 0.0 to 10.0, which can be modified by scoring the optional Temporal and Environmental metrics.

In the current version of CVSS (CVSSv3.0), which released in June 2015, the base metric group is composed by the Access Vector, Access Complexity, Privileges Required and three impact metrics composed by confidentiality, integrity, and availability. The first group captures how the vulnerability is accessed, whereas the temporal metrics include technical details of a vulnerability, the remediation status of the vulnerability, and the availability of exploit code or techniques. Finally, environmental metrics capture the characteristics of a vulnerability that are associated with a user IT environment. Although CVSS is much similar to CWSS, some metrics, like likelihood, have been removed, leading to simpler to calculate metrics.

CVSS has been widely adopted, especially the use of base scores from the Base metric group, and it represents a widely established approach. Examples of its usage is in the Common Vulnerabilities and Exposures (CVE) [31] created by MITRE and in the National Vulnerability Database (NVD) [32] created by the NIST.

3.3 Operationally Critical Threat, Asset, and Vulnerability Evaluation (OCTAVE)

The Operationally Critical Threat, Asset, and Vulnerability Evaluation [33] was proposed by the Carnegie Mellon University, United States of America. The methodology covers different aspects of risk management, including the identification of relevant assets for a certain organization, and the vulnerabilities and threats associated with such assets to define a strategy accordingly. It should be noted that OCTAVE focuses on operational risk and security practices, not on technological aspects related to the risk to be assessed. Therefore, it is not easily applicable to most vehicular systems, also considering the high amount of documentation, training and practices that are necessary to apply it.

Although OCTAVE framework defines eight processes, before performing such processes, an exploratory phase or Phase Zero is used to come up with the criteria to be used during the application of the methodology. While it defines a set of metrics, the mapping with impact intervals (low, medium, and high) is open to subjective

interpretation complex documentation, which has motivated the development of a more lightweight alternative called OCTAVE-S [34].

3.4 DREAD Algorithm

The DREAD algorithm [35] is used to compute a risk value, which is an average of five categories: Damage potential (how great is the damage if the vulnerability is exploited?), Reproducibility (how easy is it to reproduce the attack?), Exploitability (how easy is it to launch an attack?), Affected users (as a rough percentage, how many users are affected?) and Discoverability (how easy is it to find the vulnerability?).

The calculation always produces a number between 0 and 10; being the higher the number, the more serious the risk. However, there is not a consensus on how the actual risk point scale should be, since it all depends on the individuals performing the threat modelling [36]. DREAD requires scoring each of the five categories on a scale from 0 to 10, which leads to discussions on the fine differences between consecutive numbers, e.g., 5 and 6. This problem is still bigger in larger organizations with multiple teams, as reaching to an agreement could be challenging. One solution to this problem, as remarked in [37], is using scores of High, Medium, or Low, that are easy to agree, instead of using Microsoft's eleven-valued scale. For example, a simple scheme would be: High (10 points), Medium (5 points), and Low (0 points) when it comes to Damage potential, and Hard (0 points), Medium (5 points), Easy (10 points) when it comes to Reproducibility.

3.5 VERACODE

The Veracode Rating System [38] is an adaptation of CVSS to evaluate detected weaknesses/vulnerabilities. Veracode assigns a severity level to each flaw based on Confidentiality, Integrity, and Availability. Each severity level reflects the business impact if a security breach occurs in these three security aspects.

The overall Security Quality Score, based on its associated CWE entry, is computed by aggregating impact levels of all weaknesses within an application, so multilayer and aggregation are considered in this way. It ranges between 0 and 100, where 0 is insecure and 100 means that no flaw has been discovered. This score does not predict vulnerability potential as much as it enumerates the security weaknesses and their impact levels within the application code. The score calculation includes non-linear factors so that, for instance, a single Severity 5 flaw is weighted more heavily than five Severity 1 flaws and so that each additional flaw at a given severity contributes progressively less to the score. However, the score only enumerates security flaws and their impact, it is not capable of predicting potential vulnerabilities. The weights are exponential and calculated empirically by Veracode's application security experts, meaning that they can be affected by a personal judgement. Finally, the score is normalized to a scale of 0 to 100, where 100 means no flaw has been detected for the application business criticality [39].

The assurance levels follow a three-letter rating system (from A to F). The first letter is used for the results from the binary analysis, the second for automated dynamic analysis, and the third for human testing. They are used to determine the extension of

the testing (e.g. higher assurance levels could imply more testing techniques) and the overall acceptance criteria (e.g. a lower assurance level can be accepted with lower security scores if it does not pose a high business risk).

3.6 CENZIC Hailstorm Application Risk Metric (HARM)

The Cenzic HARM [40] Score is a quantitative score for the risk associated with a web application. The metric is split into 4 impact areas relevant to web application security: the browser, the session, the web application, and the server. It also considers two additional factors: a complexity factor and the precision associated with detection of a given vulnerability and a modifier called weight, which users can use to modify the obtained risk. However, this method does not account for the relationship of vulnerability properties, which are also important in the evaluation of the distribution of exploitation, and it is focused only on web applications.

Mathematically, the Base Risk Equation is $Risk = 10 * 2^I$, where *I* is the impact area value. Any vulnerability can impact a Web application in up to 4 different ways (4 impact areas). Within those 4 areas, the degree of the risk can be 1 (“low”) to 5 (“Critical”), represented as rings inside a circle. To determine the application risk level (impact value) for a vulnerability, HARM uses security values with five degrees of risk such as confidentiality or access. The vulnerability risk is the sum of the risk score from each of the four impact areas, which can be modified by the weights from other metrics (e.g., attack complexity or detection precision). Finally, the HARM rating is calculated by multiplying all the identified vulnerabilities (that can include different components and layers) within an application by the level of importance managers give to that application, so it gives the possibility of indirectly considering the context by changing the weights.

3.7 Open Web Application Security (OWASP)

The Open Web Application Security (OWASP) Risk Rating Methodology [41] is part of the OWASP project, which provides a basis for testing web application technical security controls. The risk rating methodology estimates the risk in terms of likelihood and impact following several steps. The first one consists of identifying a risk to be rated, analysing, and gathering information about it. The second step analyses factors for estimating Likelihood. It is not necessary to be over-precise in this estimate. Generally, identifying whether the likelihood is low, medium, or high is sufficient. There are several factors that can help determine the likelihood, such as the ease of discovery and exploit or the skills of the attacker. The third step is about identifying factors for estimating Impact, divided in technical impact on the application, the data it uses, and the functions it provides and in business impact on the business and company operating the application. In this sense, the context factor can be considered through this metric. The fourth step determines the risk severity. The likelihood and impact estimate are put together to calculate an overall severity for this risk, obtaining none, low, medium, high or critical. Finally, it is decided what to Fix. It is also possible to customize the Risk Rating Model, for example adding factors, customizing options, or weighting the factors.

The main limitation of OWASP is that it is only focused on web applications, domain in which there is no current standard [42]. As in the other schemes, the scale used (low,

medium, and high), based only in the consensus of the testers, make the result subjective and variable, depending on the person that is measuring the risk, and, although it is not required to have a high precision on calculating the likelihood, this is one of the most difficult metrics to calculate.

3.8 E-safety Vehicle InTrusion Protected Applications (EVITA)

The E-safety Vehicle InTrusion protected Applications (EVITA) [43] risk is evaluated based on the severity and attack probability (attack potential). The attack severity of an attack includes four different aspects that may be associated with harm to the stakeholders (operational, safety, privacy, and financial aspects) as a 4-component vector with a range of qualitative levels. The severity of an attack is assessed using attack trees, by considering the potential implications of the attack objectives for the stakeholders. The attack potential is a measure of the minimum effort to be expended in an attack to be successful, and it includes four different factors: elapsed Time to identify the vulnerability and develop the attack, specialist expertise required, knowledge of the system under investigation, window of opportunity (if access should be continuous, if online preparation is needed, etc.) and IT hardware/software or other equipment needed. The probability of a successful attack is also assessed using attack trees, by identifying combinations of possible attacks on the system assets that could contribute to an attack method.

Finally, the risk level is determined from the severity and the combined attack probability associated with a particular attack method. This is achieved by mapping the severity and attack probability to the risk using a “risk graph” approach. Towards this end, the probability and severity combinations are mapped to a series of risk levels ranging from 0 (lowest) to 6 (highest).

Although EVITA is very similar to OWASP methodology, EVITA is more tied to the assets involved through the threat agent factors (knowledge of the target and equipment required), making it more suitable for the automotive industry.

3.9 HEAling Vulnerabilities to ENhance Software Security and Safety (HEAVENS)

HEAVENS [44] defines a framework for identifying security requirements in the context of smart vehicles, representing an improvement of the EVITA methodology. Towards this end, the HEAVENS methodology identifies assets and associated threats. Threats are then mapped to security attributes, deriving a security level for each asset-threat pair. HEAVENS considers the following security attributes: confidentiality, integrity, availability, authenticity, authorization, non-repudiation, privacy, and freshness.

HEAVENS analyses threats based on Microsoft’s STRIDE approach (see BIECO D7.1), which is used for threat analysis, and ranks the threats based on a risk assessment. The security level is calculated using two metrics: threat level and impact level. Like the

EVITA methodology, the threat levels, which reflect the likelihood of the threat, are computed based on the same parameters used in Common Criteria. It uses a scale of 4 to 0, with 4 being very high and 0 being low likelihood. The scale for impact level ranges from 4 to 0 as well, with 4 being critical and 0 being Quality Management. The impact of the threats is quantified by considering the expected loss of the objectives, which are safety, finance, operation, and privacy.

HEAVENS improves EVITA in several ways. For example, EVITA does not distinguish among various access types (e.g. physical, logical) while rating opportunity and although the impact level is aligned with ISO 26262 (an international standard for functional safety), it does not provide a suitable guideline to estimate the impact and it does not take other legislation aspects into account for risk rating. In this sense, HEAVENS provides estimation of impact level parameters (safety, operational, financial, privacy and legislation) based on industry standards. For example, the safety parameter is aligned with the ISO 26262, financial parameter is based on the British Standards Institution Standard, and operational parameter is based on the Failure Mode and Effect Analysis (FMEA) proposed by the Automotive Industry Action Group (AIAG).

However, HEAVENS security model also has some limitations. It does not suggest countermeasures or security mechanisms to assist in fulfilling the derived security requirements and empirical security measurement (e.g., testing) is neither considered.

3.10 Threat, Vulnerability, and Risk Analysis (TVRA)

TVRA [45] is an assessment method developed by the European Telecommunications Standards Institute (ETSI) developed for data and telecommunication networks.

Through the TVRA method the TOE, the associated assets (physical, human or logical) and the goals of the evaluation are firstly identified. Security objectives are then identified and classified based on the five security attributes: confidentiality, integrity, availability, authenticity, and accountability. From them, it is possible to derive the functional security requirements, which are more detailed requirements than the security objectives, e.g., passwords should be used for authentication. An inventory of assets is done, and possible vulnerabilities are identified and classified along with corresponding threats and undesirable results. These threats are classified according to the following four categories: interception, manipulation, denial of service and repudiation. The risk is calculated based on the likelihood of these threats and their undesirable results. Finally, a set of countermeasures are derived, and cost-benefit analysis is carried out to select the most appropriate countermeasures to reduce the risk of the identified threats. These results are then used to design security services. However, the standard only gives the steps to be performed, not how to perform them and the process is quite large and complex.

3.11 Combined Safety & Security Risk Analysis

In this section, an overview of the state-of-the-art in safety and security analysis methods is presented in a chronological order. This work provides a complementary analysis to the threat analysis and risk assessment methods presented in [46] by looking at the characteristics of methods that perform a combined safety & security analysis from different viewpoints of the engineering process targeted towards assuring the safety and security properties of systems under development. A complete quantification of criteria together with a comparison between them can be found at [61].

Following the 2011 ISO 26262 recommendation of "reasonably foreseeable misuse" as a factor risk analysis [47], further elaborations of misuse with malicious attacks have emerged, with recommendations for security practices being published 9 years later [48]. Due to the fact that, (*a*) at the moment when security practices started to emerge, safety was well established in the automotive domain, and (*b*) security incidents can compromise safety, first security approaches have proposed a joint addressing of safety and security at all stages of system development, initially in a component-oriented fashion [49]. As reported in [50], safety is concerned with preventing accidents through identification of potential weaknesses, events, internal hazards, and potentially hazardous states, followed by identification and implementation of appropriate mitigation mechanisms to reduce the risk to a tolerable level. On the other hand, security is concerned with protecting assets against internal or external threats and vulnerabilities that can compromise them. A mechanism for protecting assets is using control strategies that reduce the risk of compromising the functionality to an acceptable level. Driven by the first approach to use a component-level perspective for combined hazard and threat analysis as a safety & security mechanism performed during concept design, popular safety analysis methods were further extended to include security analysis as well. For example, FMEA (Failure Modes and Effects Analysis) [51] was extended towards FMVEA (Failure Mode, Vulnerabilities and Effects Analysis) [52], HARA (Hazard Analysis and Risk Assessment) [53] was extended towards SAHARA (Security-Aware Hazard Analysis and Risk Assessment) [54], STPA(System Theoretic process Analysis) [55] towards STPA-Sec (Systems-Theoretic Process Analysis for Security) [56].

3.12 Combined Harm Assessment of Safety and Security for Information Systems (CHASSIS)

CHASSIS [57] was introduced in 2012 for supporting the derivation of safety and security requirements of a system. For an already developed system, design artefacts can be reused for CHASSIS as well, otherwise, if the system is in the design phase, safety and security analysis can be performed in parallel. During CHASSIS, functional requirements artefacts, such as use cases and sequence diagrams, are analysed with HAZOP (Hazard and Operability studies) keywords. The analysis results in a set of safety and security requirements documented in UML (Unified Modelling Language). During elicitation of functional requirements, D-UCs (Diagrammatic Use Cases) and T-UCs (Textual Use Cases) are used for describing users, functions, and services, whereas UML sequence diagrams are used to model sequences of interactions between objects. During the

elicitation of safety and security requirements, potential misuses of the system are identified in brainstorming sessions that involve domain experts as well as safety and security experts. Misuses are identified through a combination of use cases and HAZOP guide words, documented in D-MUC (Diagrammatic Misuse Cases), T-MUC (Textual Misuse Cases) and MUSD (Misuse sequence Diagrams). Failures are documented in FSD (Failure-Sequence Diagrams).

3.13 Failure Mode, Vulnerability and Effect Analysis Method (FMEVA)

FMVEA method [52] was introduced in 2014 as a security-oriented extension of the FMEA (Failure Mode and Effect Analysis) with the scope of unifying safety and security cause-effect analysis. Extending the analysis of a given failure mode, which describes how a system quality attribute fails, FMVEA proposes a threat mode, which describes how a security attribute of a component fails. Vulnerabilities are considered the causes for failures of security attributes. Vulnerabilities analysed together with potential attacks form the likelihood of a threat mode. Following the system decomposition into functions, the FMVEA analysis provides a combined list of failures and threat modes together with their causes and a risk estimate. In a cyber-physical system such as a vehicle, typical functions are either processing or communication functions. As such, over time, frequently encountered failure modes for input and output have been identified [58]. Performed during system design, FMVEA provides a systematic analysis of failures and threats effects, allowing development to proceed effectively. Because FMVEA focuses on the analysis on component functions and their interactions, it does not address the system-level functions explicitly. Consequently, the analysis can result in a list of vulnerabilities that can be potentially exploited for each component but may miss critical threads to the overall system.

3.14 Security-Aware Hazard Analysis and Risk Assessment (SAHARA)

SAHARA technique has been introduced in 2015 [54] as a joint approach for performing safety and security analysis through a combination of HARA [20] for the automotive domain and STRIDE (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privileges). Focusing on the automotive domain, SAHARA quantifies the probability of an identified security threat and aligns its Security Level (SecL) to the ASIL (Automotive Safety Integrity Level) defined in the HARA. As in CHASSIS, functional use cases are analysed but in SAHARA, the STRIDE model is used for the identification of possible security threats. Latter, Macher et al [59] recommend using DREAD (Damage potential, Reproducibility, Exploitability, Affected Users, Discoverability) model instead of STRIDE for distributed systems as an approach that performs concern-oriented evaluation of safety considerations influenced by security aspects for distributed systems within the automotive domain.

3.15 System Theoretic Process Analysis - Security Method (STPA-Sec)

STPA-Sec [56] is an extension of the base STPA approach based on the STAMP (System Theoretic Accident Model and Processes) causality model. Starting from the idea that

an attacker could infiltrate a system by exploring an open and undiscovered vulnerability, once the intention is there, Young and Leveson introduce the idea that the goal of security should be the assurance of maintainability of critical functions despite such intrusions [60]. Therefore, the STPA-Sec, unlike other Focusing on enforcing safety and security concerns at the system level, hierarchical control structures are created where higher-level components constrain lower-level components. As a result, the safety and security incidents are treated as a hierarchical control problem rather than a simple chain of events. STPA-Sec works on a specification of the system accidents and losses, along with hazards that can potentially lead to accidents. The method performs a systematic analysis according to existing functional control diagrams. Then, a set of guide words are used to reason about causes of hazards.

For further comparison on these approaches according to three different criteria: “Type of Approach”, “Needed Artefacts”, “Creativity of Analysis process”, “Application Phase”, “Generated Artefacts”, “Risk Evaluation”, “Usability” and “Efforts” within a use case can be found in the BIECO publication “Comparison of Safety and Security Analysis Techniques” [61].

4. Security Testing Analysis

The next subsections review the state of the art of the main testing techniques that could be used to empirically validate the security offered by a system.

4.1 Model Based Testing (MBT)

MBT [62] is a testing approach based on the usage of models that can represent the test requirements, the System Under Test (SUT) itself, and its environment. In [63] is the MBT variant that validates the security behaviour of a certain software system requirements considering a set of security properties.

The model is typically interpreted as a high-level view of the SUT, with high level operations (e.g., sendMsg(valid_id)), and high-level types (e.g. ID_TYPE). Therefore, an additional process to turn the abstract model into a specific implementation of the SUT is required. To this end, the technique employs an adapter, which serves as an interface for implementing the real tests that will be run on the SUT. This adapter is mainly composed by two different interfaces: one for the high-level operations and one for the high-level types. The interfaces, that are usually written in the testing language used (e.g. Java, TTCN3), should be implemented by the tester, linking each high-level operation with a specific command/message on the TOE and each high-level type with a specific real value. Therefore, the test suite calls the adapter, which translates the operations and types of each test step and sends it to the TOE to execute the test step.

The following steps are normally included in the MBT (and Model-Based Testing, MBST) process:

- Model Design: in this first step, the high-level model of the SUT is designed and created from specifications that will drive the specification of testing criteria to evaluate the SUT. Towards this end, formal languages such as UML [64] and proprietary languages or Domain Specific Languages (DSL) [65] can be used. The model also specifies the entities and operations that comprise the SUT, as well as their corresponding inputs and outputs.
- Test Specification: using the previous model, the next step is to define in a high-level way the tests to be implemented.
- Test Generation: the model and the previous test specification are used to generate abstract test cases that make use of the high-level operations defined in the model.
- Test translation: The tests are created by transforming the abstract tests that were previously generated into runnable tests using adapters. Towards this end, a mapping is created between the operations specified in the high-level model and the real operations of the SUT.
- Execution of the tests: the adapted tests are executed manually or automatically.

One benefit of MBT and MBST is the ability to generate high-level tests automatically. Indeed, a large number of MBT-based tools [66], such as CertifyIt [67], Graphwalker [68] or MISTA [69], are currently available to provide such an automated approach.

There are two main viewpoints based on MBT. On the one hand, Behavioural MBT [62] focuses on modelling the behaviour of the SUT by defining the operations and entities involved. The correct behaviour is modelled in this way, and it is then used to create test cases or to check for misbehaviour at run-time [70]. Furthermore, if the model is focused on modelling a malicious attack, it is called Attack patterns MBT [71]. In this approach, the pattern describes the target, individual behaviour, conditions, and postconditions of the attack to be implemented. Another way to categorise MBT approaches can be seen in [72], based on two dimensions: automated test generation (how much of the system is modelled) and risk (integrated or not in the model). Finally, when considering the generation of tests, on-line testing prevents the generation of all tests prior to execution, reducing the potentially large number of tests. Furthermore, the results of previously executed experiments may be used to direct the development of each test.

4.2 Penetration testing

Penetration testing [73] is a testing mechanism to identifying methods for circumventing a SUT's security features by simulating real-world attacks. While this method is usually used to test the SUT's missing functionality or side effects, it may also be used to test the system's environment (e.g., by exploiting an obsolete operating system). Penetration testing may be classified as black-box if the intruder has minimal knowledge about the system, or white-box if internal details are known.

Furthermore, though manual penetration testing is common, there are tools that assist the tester in discovering flaws in a more automated way. This is the case of port or vulnerability scanners [74], which use different techniques to detect security issues in applications. The scanner is used to run a series of predefined attacks against the system interfaces. The system responses are then evaluated to see whether the attack was effective, or, at least, if it was useful to learn more about it and make the attack successful.

There are many penetration testing standards; in this regard, the Open-Source Security Testing Methodology Manual (OSSTMM) [75] is the most prominent approach, which offers rules and guidelines for covering the different protocol stack layers. From initial requirements review to the report generation, the OSSTMM approach encompasses the entire risk assessment process involved in a penetration test.

4.3 Fuzzing Testing and Dynamic Taint

The concept of fuzzing testing [76] is to test SUT security vulnerabilities by using unintended or incorrect inputs. This method has been shown to be successful in identifying weaknesses that are missed by other testing methods. On the one side, it can be used to test input data (data fuzzing testing [77]) by feeding the SUT with random data to find possible errors or vulnerabilities. On the other hand, it can be used as behavioral fuzzing testing [78][79], in which valid/invalid message sequences are used with the same purpose.

Since fuzzing testing does not need knowledge of the SUT implementation details, it is referred to as a black-box testing process. Behavioral fuzzing testing, like in MBT and

penetration testing, can be on-line or off-line, depending on the test generation. Regarding data fuzzing testing, the followed approach is typically off-line, since tests are generated before they are executed.

A fuzzing technique is based on a fuzz generator, also known as fuzzer, which is the algorithm in charge of randomizing using different strategies. The most basic form of fuzzing testing is the random fuzzing, in which data for testing is randomly generated. Other more sophisticated methods are the mutation-based fuzzing testing [80], the model-based mutation testing [81], the model inference algorithm (MIA), supported evolutionary fuzzing testing [82], or the method suggested by [83], which combines MBT and fuzzing techniques.

In mutation-based fuzzing, the fuzzer has some knowledge of the SUT's input format, so the mutation-based fuzzing tool will create new variants based on existing data samples (mutants). In the model-based mutation testing approach, the attack model is mutated by choosing different attack paths. Finally, MIA evolutionary fuzzing incorporates model inference and a genetic algorithm to exploit a SUT's possible cross-site scripting vulnerability. Since the tests are generated based on the attacker model (on-line or off-line), this technique is similar to MBT. Despite the high efficacy of this technique and its ability to detect zero-day vulnerabilities, each SUT needs its own fuzzer. Furthermore, the number of generated tests must be limited by a specific parameter or property. These factors may pose significant scalability challenges, especially in complex systems with a wide number of components, as in the vehicular one.

To enhance fuzzing testing, the aim of dynamic taint [84] is to label specific data (e.g., coming from an untrusted source) as tainted. Then, the way the SUT employs the tainted data can be used to recognize insecure data flow. Unlike static analysis, which focuses on identifying problematic data flows, dynamic taint analysis is carried out in the background while the SUT is running. This technique can be combined with fuzzing testing to gain information about path execution by choosing the most promising test sequences to detect possible vulnerabilities [85].

4.4 Regression Testing

Regression Testing [86] is mainly used to test changes over the SUT, verifying that there are no collateral effects, and it provides the expected functionality correctly. Therefore, when the product is updated or patched or when the product has been modified due to new requirements, regression testing is necessary. In security, Regression testing is especially relevant when a new vulnerability is discovered and/or when a security patch is needed to solve a security issue.

There are five main regression testing approach [72] depending on the deep of the testing process: test all, reduction, minimization, prioritization, and selection. The first one, test all, consists of repeating all the tests completely. This is the least efficient approach, as it could involve an expensive and time-consuming process. The rest of the approaches try to reduce the number of tests executed, selecting a subset of relevant tests to perform the process in an efficient way. Minimization consists in reducing the testing coverage, by removing some tests that are not relevant for the changes. Prioritization orders following specific criteria, in a way the more relevant tests are

executed in first place. Finally, selection select a subset of tests directly involved with the changes made over the product.

There are tools that facilitate the automation of the regression testing. In this sense, [87] represents an extension of the tool CertifyIt mentioned before in MBT, to support regression testing.

4.5 Code-Based Testing

Code-based testing [88] is a white-box technique to detect vulnerabilities and faults by looking at the source code, to detect anomalies at the very early stages of the development. Code reviews can either be done manually, in which an expert is reading program code line-by-line [89], or automated that is usually called Static Code Analysis (SCA) or Static Application Security Testing (SAST) [90]. In SAST, a tool reviews the application code and automatically reports potential security flaws. It can use syntactic checks such as calling insecure API functions or using insecure configuration options. It is also possible to use semantic checks that require an understanding of the program semantics, such as the data and control flows.

While manual code reviews are a tedious process that requires skill, experience, and persistence, SAST tools can analyse all control flows of a code in a scalable way. Based on that, these tools can provide detailed recommendations to fix security issues very early in the development process. Compared to dynamic test approaches, SAST tools provide a higher coverage of the SUT and a lower false negative rate, that is, when the threat exists but the test passes. However, these tools only report known vulnerabilities, and, therefore, an expert is needed to configure the tool and identify such vulnerabilities.

4.6 Combinatorial Testing

Combinatorial testing is a black-box testing technique which relies only on the input/output behaviour of the system. More precisely, in combinatorial testing, test cases are designed to execute combinations of input parameters. Providing all combinations is usually not feasible in practice, due to their extremely large numbers. Dealing with this problem, combinatorial approaches able to generate smaller test suites, for which all combinations of the features are guaranteed, are preferred. Among them, common approach is all-pair testing technique, focusing on all possible discrete combinations of each pair of input parameters.

4.7 Mutation Testing

Mutation testing is a technique in which syntactic faults, simulating typical programmer mistakes, are seeded in the original program in order to produce a set of faulty programs, called mutants, each one containing one fault. The main purpose of mutation testing is to assess the adequacy of a given test suite. Each test case is executed on the original program (also called Gold program) and its mutants; then, outputs are collected: if the mutant output is different from the original program's one, the fault is detected and the mutant is said to be killed. The mutation score is the ratio of the number of detected

faults over the total number of seeded faults and indicates the effectiveness of the test suite. Since mutation testing was proposed in the ’70s, it has been applied to many programming languages, such as Java, Fortran, Ada, C, SQL, and many mutation tools have been developed to support automated mutation analysis. We refer to [91] for an extensive survey of software mutation testing. The general process of mutation analysis consists of two steps: first, change the original program with predefined mutation operators and generate a set of mutated programs, called mutants; then, the mutants are executed against a test suite and information is collected during the execution for various purposes of analysis.

5. Challenges of Security Certification and Evaluation

This section reviews the main challenges approaches for evaluating the security of a specific TOE, which have been obtained from the analysis of current certification schemes, risk assessment mechanisms and testing techniques.

5.1 Heterogeneity

One of the main challenges associated with security certification is the harmonization of the wide variety of security certification schemes that coexist together [22]. The current heterogeneity makes difficult the comparison of different solutions and processes, especially when a product is evaluated under different certification schemes at national levels. Currently, there is no unified solution that copes with these issues. Therefore, the process of comparing and assessing the cybersecurity level of different products is challenging. ENISA already remarked the need for harmonization of security certification at least at the European level, which could help to increase the trustworthiness and competitiveness of European products [92]. Following the recommendations of ENISA, there are some elements that should be harmonized. This is the case of the different assurance levels, the elements considered during the certification process and the roles of the involved stakeholders. Regulatory bodies have an important role here, promoting the creation of a cybersecurity framework through the consensus of the main stakeholders and orchestrating its development and deployment. In particular, the certification meta-scheme proposed by ECSO [93] represents an ambitious initiative homogenise and aggregate different certification approaches under a common framework.

5.2 Dynamicity

Cybersecurity is a very dynamic concept. Considering the frequency of updates and patches of certain devices, a lightweight recertification process is necessary to ensure an updated security certificate. Automated procedures are also necessary to ensure the scalability of the (re)certification process. In this sense, the cybersecurity certification scheme should deal with the changes of the certificate. On the one hand, the product should be monitored during its lifecycle to detect new vulnerabilities and update its security level. On the other hand, the security level should be modified when a cybersecurity recertification is required due to an update/patching.

5.3 Cost and Time

The existing approaches for cybersecurity assessment (e.g. CC, OCTAVE, EVITA) are oftentimes consuming and complex, requiring formal documentation and processes [94][95] which could imply that such process can impact the launch of a new product. From the certification point of view, a costly process could imply that the manufacturer cannot afford the costs related to the cybersecurity certification process. On the one hand, the cybersecurity certification could require monetary costs related to the payment to a certification body issuing the certificate, or the testing laboratory being involved in the process. Furthermore, the process can involve qualified personnel to implement the

measures required to obtain the certification. This issue is even more relevant in case of update, which may require a re-evaluation and re-certification.

Based on this, the cybersecurity evaluation process should be cost-effective and lightweight to foster its adoption, facing the trade-off between EAL and the costs for the companies (time and money), especially for SMEs and startups. It could also help to deal with security changes, by providing a faster and affordable re-evaluation process.

5.4 Objectivity, Reproducibility and Repeatability

Common risk assessment approaches are based on the use of different security metrics, which are employed to provide a more reliable assessment of a product security level. However, some of these metrics, such as likelihood, are difficult to be measured as it requires historical data or the opinion of an expert. Consequently, the assessment process involves a certain degree of subjectivity depending on the security expert who is analyzing the system. Indeed, some risk assessment schemes (e.g., CWSS) decided to ignore some of these metrics. This fact makes it difficult to compare the security acquired between different systems, since although the same scheme is used, different systems may have evaluated it very differently. In this sense, there is a need to support the security evaluation of the system with objective metrics derived from empirical observations.

5.5 Labelling

As a result of the certification process, a label should be generated to provide a simple, clear, and visual level of the security certified [96]. Companies such as Bosch [97], add that customers need to compare the security of different products without feeling overwhelmed with technical details. In this sense, the label must address an important trade-off between the simplicity of the label and the non-ambiguous and complete representation of the results of such process. This is rather difficult, because in comparison to the energy label, which measures a physical quantity, the measurements of security are far more complex. There are already some efforts towards the design of this visual and simple label, such as the one proposed by the Carnegie Mellon University for IoT devices [98] or the recent European Cybersecurity label of ECSO [99] focused on providing a seal for European-made products.

In addition, the label design should also take into consideration the dynamicity of the security. As pointed out by ECSO WG1 [3] and EUCC, a visual static cybersecurity label is not enough since it should also cope with the dynamism of security threats and context to reflect changes in the current security level. For this reason, the usage of a digital QR code, which can be regenerated, can help to check the status of the cybersecurity label in a fast and easy way, as it could also be easy to update.

5.6 Context

The context in which the product will operate must be considered, to make products comparable among each other and to specify the boundary conditions of the context

where the cybersecurity certification was applied. This aspect is specially challenging because it could not be known a priori.

To address the issue of label significance and the need to measure the security properties, security metrics must be established. However, some of such metrics, such as likelihood or impact, are difficult to be measured, due to its complexity.

5.7 Dependencies

A system is composed of several components performing different functionality, and moreover, the heavily-tiered ecosystem of the supply chain also leads to security integration issues. The wide heterogeneity of devices and technologies (e.g. machine learning, artificial intelligence, cloud computing, networks, sensors), that can be part of the ecosystem, derives on a heterogeneous environment that hardens the security evaluation. Therefore, it is crucial to detect cascade effects and assess the real security level of the system. It means that the risk assessment of a certain system will depend on the security level associated to each part of the system. For this aspect, the main challenge is related to the way in which each risk value could be aggregated to provide a reliable value for the whole system [100]. While most of the risk assessment schemes already consider the use of weights associated to different metrics to assess a certain product, these aggregation aspects are usually ignored. There is the need for more sophisticated security evaluation mechanisms that can capture components interdependence and cascade-effects among all the involved components. These mechanisms will help capture how interdependencies operate and will heighten impacts in order to develop procedures and policies to improve recovery.

5.8 Standardization

Another challenge is related with the standardization of the cybersecurity scheme. Despite the limitations of the current approaches, a cybersecurity certification scheme should adopt the main concepts, terms and operational aspects of existing standard approaches (such as Common Criteria (CC) [7]). The usage of standardized concepts helps to achieve a common understanding and the harmonization of the cybersecurity certification.

5.9 Transparency

From a cybersecurity certification perspective, it should be possible to reuse the cybersecurity certificates of such components for the cybersecurity certification of the system as a whole. Whereas reusing the certificates helps to obtain a more cost-effective certification of the system, some questions arise related to the certification information to be shared, and how it should be disseminated. Specifically, a trade-off should be established between the visibility of the cybersecurity certification data (processes, tests, vulnerabilities, risk level etc.) and the right of the certification body to protect such data. Furthermore, the development of a software component may not be linked to a specific product or system. Thus, the certification of this module in a certain hardware and operating system may not be valid for the composition of a specific

system. This aspect could hinder the potential reuse of previous certifications to be used for the certification of the whole system. In this case, it is important to identify which information from the evaluation process could help to avoid (at least partially) the re-evaluation of that component. If proper actions are not in place, a new certification process could be required with an additional effort and cost in the system certification.

In this regard, a European database containing all the information related with cybersecurity certification (e.g. test reports, risk level) would help to come up with a more harmonized cybersecurity certification composition approach. Furthermore, this database could also provide transparency by giving details about the certification process itself.

5.10 Security Disconnected from Safety

As mentioned before, a key challenge is the growing interconnection of the systems. Due to this connectivity, it is no longer acceptable to think that safety-critical domains, such as the automotive one, are immune to security risks. Traditionally, safety and security have been considered separately, but due to the growing recognition of the mutual impacts, a combined view becomes more important. The key point is that cybersecurity, privacy, and physical safety can no longer be treated as separate concerns. When attackers can affect the physical operation of the system, network cybersecurity and physical safety become interdependent. Therefore, risk assessment processes should also address security concerns, defining security validation processes that explicitly address abuse cases and attacks.

In domains where safety is crucial, people are usually skilled and used to deal with safety issues during the development of the system, mainly due to the existence of safety standards well established years ago. However, when taking about security, there is a lack of knowledge on how to deal with protection of the system. In particular, the automotive industry mostly assesses security with the same methods as safety (following methods similar to ISO 26262). These standards marginally address security and help reducing malfunctions and failures (random and systematic faults) but do not protect against attacks.

5.11 Lifecycle Management

As mentioned above, security is a very dynamic concept. A system that has already been evaluated may be subject to threats, either due to deficiencies in the evaluation process or due to threats that are not yet known, that is, zero-day threats. This not only involves the evaluation and the need for a subsequent re-evaluation in case a vulnerability is detected but it also makes it necessary for this process to be part of a security management framework that addresses the entire life cycle of the system. In this sense, a crucial aspect is the integration of monitoring techniques and tools to detect potential security attacks or threats in a timely way. In addition, for a correct management of the security during the life of the system it is necessary to obtain information about the new vulnerabilities. Information sharing between the different manufacturers and entities is also a crucial aspect that is not always achieved. Most of the relevant knowledge lies

with suppliers but component suppliers do not usually share component specific information, so there is information asymmetry between manufacturers and suppliers.

5.12 Privacy Certification

Cybersecurity does provide controls for data protection, but organisations will require additional considerations to satisfy privacy goals.

Considering the data privacy aspects, for protecting “personal data by-design” and gaining legal compliance with the GDPR, several challenges have to be faced, such as:

- **Performing Data Protection Impact Assessment.** Performing Data Protection Impact Assessment (DPIA) in accordance with the GDPR is pivotal to promote and achieve privacy-by-design. For the different organizations, fulfilling the GDPR requirements is an integral part of their business. The challenge here is that the GDPR’s requirements are often too vague and open. This makes them subject to interpretation. Therefore, it might be difficult to comply with them correctly and completely.
- **GDPR-based development life cycle.** The available development life cycles do not completely incorporate the privacy-by-design principles, and proposals targeting the GDPR’s demands are still needed. Therefore, a reference GDPR-based development life cycle for the specification, implementation and testing of software systems and applications which considers (European) legal requirements is needed.
- **Enforcing and demonstrating the privacy principles compliance.** The peculiarities and the complexity of the currently available systems and applications call for specific automatic approaches, facilities, and tools for enforcing and demonstrating the privacy principles compliance. This is a crucial aspect for the successful and lawful privacy-by-design process development.

6. Proposed Security Certification Methodology

As explicitly mentioned in the Cybersecurity Act [101], *A European cybersecurity certificate [..] shall provide assurance that the ICT products, ICT services and ICT processes for which that certificate is issued meet the corresponding security requirements, including security functionalities, and that they have been evaluated at a level intended to minimize the known cybersecurity risks, and the risk of incidents and cyberattacks carried out by actors with limited skills and resources. The evaluation activities to be undertaken shall include at least the following: a review to demonstrate the absence of publicly known vulnerabilities and testing to demonstrate that the ICT products, ICT services or ICT processes correctly implement the necessary security functionalities*.

The proposed methodology for security evaluation and certification explicitly considers these processes and deals with some of the challenges analysed in the previous section. In particular, the methodology follows a test-based risk assessment approach, in which the risk assessment is improved by a security testing process. Security assessment and testing have been already considered in current literature as essential processes for cybersecurity evaluation. Indeed, ECSO considers them as key elements for cybersecurity certification: *It would be convenient to consider a security testing methodology (to) help in [...] the process of updating the certificate in a fast, easy and inexpensive manner. When doing an update or patch, security tests can be executed to assist reassessment processes...[93]*.

UNECE regulation for smart vehicles considers that the processes used within the manufacturer organization to manage cybersecurity should include activities to identify, assess, categorise, test and treat risks, as well as activities to monitor, detect and respond to cyber-attacks, cyber threats and vulnerabilities. ETSI [102] also considers the combination of risk assessment and testing defining two different approaches based on the ISO 31000 standard for **Risk Management** and the ISO 29119 standard for **Security Testing**. Furthermore, European projects such as ARMOUR or RASEN also followed this combined approach.

In the same way, the proposed security certification methodology builds a framework on top of the two main streams of this proposal: security testing to identify security vulnerabilities and security risk assessment to measure the associated risk. Figure 1 shows a high-level view of the proposed methodology, distinguishing several processes. The first one, is the **context** phase, which considers the existing regulation, the best practices, current standards etc. to build an initial set of security claims that can be used as starting point for the security evaluation. From this initial set and taking into account the particular Target of Evaluation (TOE), in the **risk identification** phase, a set of applicable threats can be selected. This set can be also extended with specific threats not considered in the initial set, by examining the special characteristics of the TOE. Once the threats are selected, the **test implementation** phase deals with the design and implementation of the tests necessary to verify if the system is vulnerable to these threats. The tests are executed in the **test execution** phase, generating at the end of the process a test report. The test results of the previous phase are used to estimate the risk of every component of the TOE during the **risk estimation** phase. Towards this end, information from the risk identification phase is required, regarding the components, the identified threats, and their impact. The **overall risk evaluation** phase combines the risk coming from every component, obtaining an overall measure of the system security. At the end of the evaluation process, a report with the evaluation results is generated. Other

actions are also possible to mitigate the security flaws encountered during the process. Additionally, the methodology also considers a transversal and supportive process for **continuous communication and monitoring** meant to deal with the lifecycle management of the TOE.

Next subsections detail the different processes of the methodology.

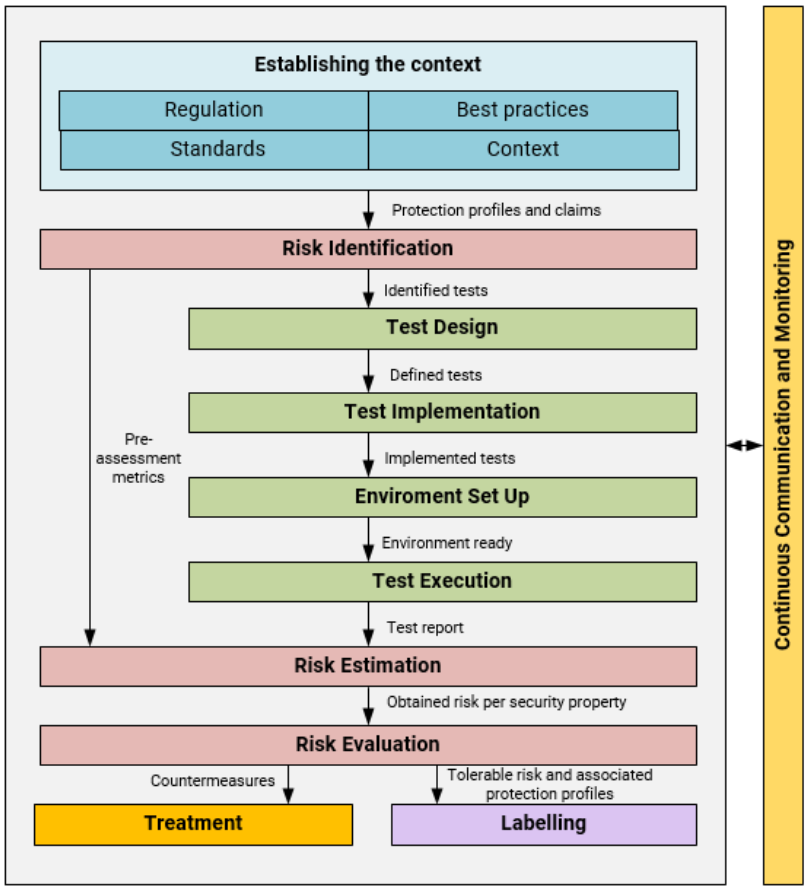

Figure 1 Overview of the security certification methodology

6.1 Establishing the Context

As already mentioned, the security evaluation proposal intends to evaluate the security level of a certain TOE. Although the TOE is defined by CC as a set of software, firmware and/or hardware possibly accompanied by guidance, we also consider its configuration (i.e., a specific protocol, libraries, cryptographic parameters etc.) as part of the TOE, as well as the context in which it is intended to operate.

Also, in order to evaluate the security of a system, a starting point is necessary, through which to analyse possible security flaws. In general, risk assessment schemes use existing vulnerabilities for this, which are collected in databases such as the well-known National Vulnerability Database (NVD) [32] as well as the security expert analysis. The starting point of the methodology defined in this deliverable is set up in the **Establishing the context** phase. Here it is not only considered known threats from vulnerability databases, but we also consider best practices and standards that could set up a basic

set of claims to protect the system against unknown threats. In this sense, D7.1 represents this starting point for the methodology of possible claims obtained from relevant sources. It is worth noting that this general set is intended to facilitate the security evaluation, but it should be complementary to more specific claims of the TOE, which are considered in the next phase.

Furthermore, this initial and general set can be used in the certification process to define a set of profiles associated to the level of testing performed or to the level of verification of the requirements, similarly to the EALs defined in the Cybersecurity Act, in Common Criteria, the Trust Assurance Levels (TAL) defined by the Car 2 Car Communication Consortium (C2C-CC), the security levels of the HEAVENS approach or the Automotive Safety Integrity Level (ASIL) levels defined within the ISO 26262 standard.

Finally, a set of Tolerance Profiles (TP) can be created, similarly to the PPs considered in CC to define the tolerance risk of the ICT in each security aspect for a specific context, as the required security may vary depending on the context in which the security of the system is being evaluated (e.g., the security needed in a car versus the security needed in an IP cam). Figure 2 shows an example of TP for a smart vehicle (transportation context). There, the tolerable risk to obtain a certificate for this product is indicated, e.g., from 0 to 7 in confidentiality risk. If a risk higher than 7 in confidentiality is obtained, the smart car cannot be certified. This is specified for each one of the 6 security properties being considered as the basis for the evaluation process (see risk identification phase, section 0). Then, the acceptable risk range is split into small sets that represent the different security levels, from A (most secure) to D (least secure), following a similar notation to the energy efficiency. This security level code will be later used to create the visual label (section 6.6).

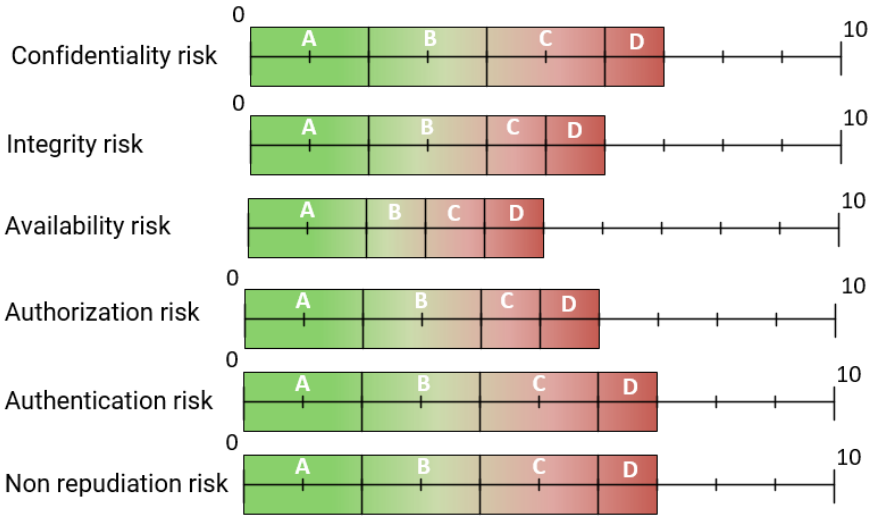

Figure 2 Example of a tolerance profile for a smart car

6.2 Risk Identification

Once an initial set of requirements and threats have been set up, the next step is to identify which risks are applicable and which are specific to the scenario and therefore have not been considered in the general set. In the **Risk identification** phase, the system is described, the components, the security properties and claims applicable to them are

identified, and, finally, the tests that can be executed against the components to validate if claims are fulfilled. Therefore, this process is intended to go from a high-level view of the whole system to fine-grained concepts such as the security tests. Moreover, Risk identification represents a crucial phase for the development of subsequent ones, since it is in charge of identifying possible risks and planning the security evaluation of the system based on this identification.
Figure 3 describes the scheme that is proposed to identify possible risks, going from the highest and abstract level, the system, to the lowest-level items that will be used in the subsequent phases, the tests. The process is the following:

- Identify the security properties that can be applicable to the system, taking into account the STRIDE (positive) taxonomy (authentication, authorization, non-repudiation, confidentiality, integrity, availability). The usage of this taxonomy will also facilitate the creation of a visual label at the end of the process since the security level of the system for each category will be obtained. It is worth noting that a system will be linked to at least one security property but not necessarily to all of them. For example, if a system does not provide resources to other systems or entities, authorization property could be not applicable for it.
- Identify components that are part of the system and that affect each security property defined before. A system can be composed of one (single-device) or more components, and some components may be linked only to a specific security property.
- Select the applicable claims for each component and security property. The set of claims are the ones predefined in D7.1, which have been obtained from relevant standards and sources, complemented with specific claims of the system, added by the security expert. The claims in D7.1 were already classified following the STRIDE taxonomy, facilitating this process to the security evaluator.
- Some of the claims defined in D7.1 are linked with the existence of known and already identified vulnerabilities in the system (e.g., C32: The source code must not use components with known vulnerabilities). The security evaluator can indicate which vulnerabilities could be applicable to each component inside these claims (*vulnerability-based claims*). The identified vulnerabilities can be obtained from public databases such as the NVD, CVE or CWE. Moreover, as these vulnerabilities are already identified, they are linked with a predefined risk value from 0 to 10 (e.g., CVSS) that can be also refined by the security expert.
- Finally, link vulnerabilities and claims with security tests. If the claim is related to the presence of known vulnerabilities (*vulnerability-based claims*), one or more tests will be associated with each of them, willing to validate if the vulnerability is present or not in the component. For the rest of the claims (*test-based claims*), identify the set of tests relevant to validate the fulfilment of the claim.

At the end of the process, there is a tree defining the system security, in which leaves represent the security tests that should be designed in the next phase. Table 1 shows an example of this output. It should be noted that, taking into account both known vulnerabilities and generic claims, one should ensure that the system is not vulnerable against known attacks but also implement additional protections in case of zero days vulnerabilities.

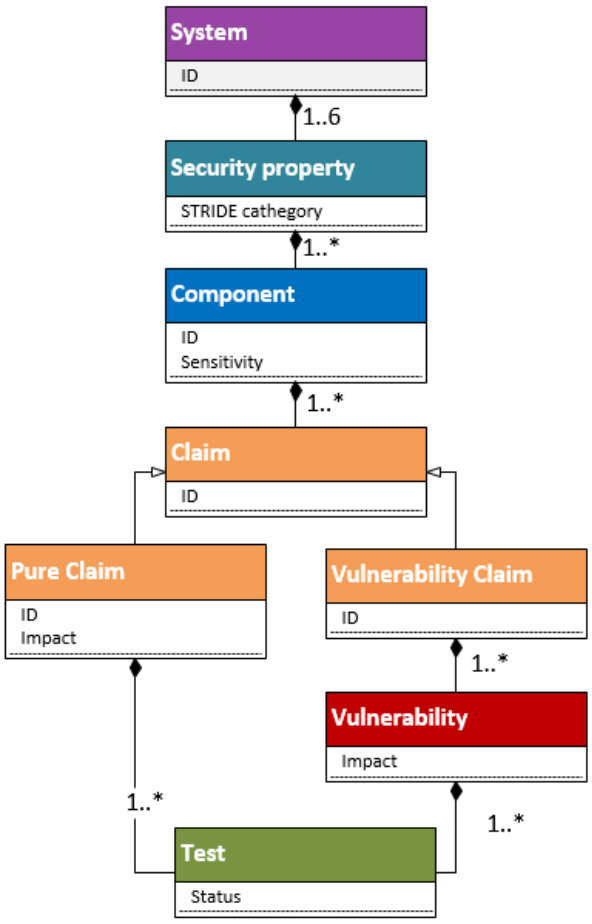

Figure 3 Decomposition of the system in the risk identification phase

This phase is also in charge of analysing the dependencies between components in order to detect cascade effects and very sensitive components within the system, that is, components whose failure could cause a failure of the whole system. A sensitivity metric between 0 and 10 will be therefore assigned to each component of the system. The more sensitive, the more prone to derive of cascade effects. This metric can be obtained manually or semiautomatically by analysing the dependency tree among the components, an idea that is going to be explored in T3.4.

It is worth noting that the sensitivity will be key to determining the overall risk of the system, but it can be also useful within the security evaluation or the testing process, with the objective of prioritising the components or the verification of the threats that can have a major impact over the system. Ensuring that a system is 100% secure is too costly and not quick enough when switching from the theory to the practice. There is a trade-off between security and applicability and, therefore, prioritisation is a powerful tool to guarantee a high percentage of security.

Table 1 Partial example of the risk identification phase output

System: Vehicle				
LEVEL 1	**LEVEL 2**	**LEVEL 3**	**LEVEL 4**	**LEVEL5**
Security property (STRIDE)	**Component (Sensitivity)**	**Claims (D7.1)**	**Vulnerabilities and Impact (for applicable claims[1])**	**Tests**
Confidentiality	Multimedia ECU (7)	C0		TEST_00, TEST_07
		C1		TEST_01
	Controller (9)	C11		TEST_02, TEST_03
		C32	CVE-2021-36988 / 7.5	TEST_04, TEST_05, TEST_06
Integrity	Controller (9)	...		
	Database (8)	...		
	OBD (6)	...		
	...			

6.3 Test Design

This task is focused on the description of the tests. Whereas test design receives as input the identified claims to be evaluated for the TOE during the risk identification phase, test design phase details them and generates an output a skeleton of the tests that should be implemented in the next task.

The identification and prioritisation of the tests should be done based on the risks that have been considered. Each test (or set of correlated tests) should be described indicating:

- A unique identification of the test,
- A brief description of the test purpose,
- Requirements needed to perform this test on the TOE,

[1] Claims related with the existence of known identified vulnerabilities from databases, e.g. claims C32, C43

- Dependencies on other tests, that is, if the execution of a test depends on the execution of a previous one (e.g. test the strength of cipher suites after testing and validating that ciphering is used),
- Inputs of the test,
- Outputs given by the test,
- Metrics that can be obtained from the test to evaluate the grade of success,
- Conditions to PASS or FAIL the test, and
- Set of steps that compose the test.

Based on that information, the most suitable testing technique and tool should be chosen. Whereas the MBT approach is highly recommended due to the automation of the testing process, this technique can be complemented with others such as fuzzing for testing inputs and random behaviours, for example. Indeed, T7.3 will provide an instantiation of the methodology combining several of these techniques. It is worth noting that different testing techniques would require additional information. For MBT, the test design phase has to address also the modelling of the TOE, including its corresponding operations, inputs and outputs. Using the previous model, the purpose of the test based on different strategies (coverage, random, requirements) can be defined, and generate the skeleton of the high-level tests. That's the main advantage of using the MBT technique, as instead of specifying test by test, step by step, all the skeleton is generated automatically from the high-level model of the TOE. How to link the high-level tests with the real system is, indeed, part of the next phase, the test implementation. ARMOUR project [103] represents an example of how MBT can be used to automate the test generation. Other examples also show how to combine and automate fuzzing testing with MBT to test the inputs and behaviours [104]. In this case, during this phase it will be required to specify the inputs and to select an adequate fuzzer.

Test Implementation

Once the skeleton of the tests is obtained, the next step is to translate them in specific low-level steps that the TOE could understand and process. The implementation of the tests can be done in a multipurpose programming language such as Python, C or Java with testing support (e.g., the JUnit framework) or in a dedicated language such as Testing and Test Control Notation version 3 (TTCN3), that is supported by TITAN.

Although the implementation of the tests is a step that requires human intervention, it can be partially automated following an MBT approach. In this case, from the model we can generate not only the skeleton of the high-level tests but also an adapter (interface) to link each high-level operation of the model with a specific operation within the real system. Although the implementation of an adapter is required, further modifications, additions and repetitions of the tests do not require significant changes of the adapter, so the re-evaluation processes become more efficient. This property is highly important in the very complex systems, due to the number of components and dependencies among them that could exist.

Environment Set Up

Once the tests are implemented and linked with the TOE, the next step is to select and prepare the environment in which the tests will be executed. The environment can be

local, using our own system and devices, or remote, making use of platforms and facilities that allow to upload the code to them, for example FIT-IoT [105], which has about 200 nodes.

In both cases we must arrange and configure certain aspects. In the local option, we have to select the devices that are needed for the testing execution, upload the code, configure it and, if we use MBT with adapters, make the necessary changes in the code to link the adapter with the real system. In the remote option, it is also required to select the appropriate remote devices that will be part of the testing, additional configuration, or entities (e.g., a sniffer), the time we need these devices and upload the code. Although the local option gives us more control of the testing process, the remote one also offers some benefits, such as the possibility of executing tests that require a high number of support entities without additional cost (e.g., DoS attacks, complex scenarios, highly dependent systems, etc.).

Test Execution

After we have the environment and the TOE ready to be tested, the tests can be executed. To automate this process as much as possible, it is highly recommended the usage of testing platforms such as JUnit/Eclipse (for Java tests) or TITAN (for TTCN3 tests), which is a compilation and execution environment for different platforms. In an MBT approach, the adapter is placed between the testing environment and the testing execution platform, acting as a translator of the different test steps.

Test Report

At the end of the test execution phase, a test report is generated to collect the results of the testing process. The result of the test is given based on the test specification performed in the test design phase. Therefore, three possible values can be obtained:

- PASS, if the test result meets the conditions of the tests specification.
- FAIL, if the test result does not meet the conditions for passing the test.
- Specific metrics, which are not exclusive with one of the other values PASS/FAIL. These metrics provide additional information: more refined than a binary result, such as the encryption percentage, the algorithm used or the length of the key. These numerical values help to improve and have a more accurate risk estimation.

6.4 Risk Estimation

The results of the tests (i.e., the test report) are then used during risk estimation to establish the security level in a more refined and objective way. Although the STRIDE approach is followed to classify the different security properties, the associated DREAD model, as a candidate of risk assessment, is not considered. The main reason is the subjectivity and inconsistency of the risk rating, which is why Microsoft abandoned this model in 2008 [106]. Hence, STRIDE but not DREAD for risk assessment is considered, and the well-known formula [107] to measure risk by combining the likelihood of exploiting the vulnerability and the impact if this vulnerability is exploited, which is also used in BIECO T6.2:

*Risk=Likelihood*Impact*

Even if the two factors of the equation have been refined in different risk assessment schemes (e.g., CWSS, CVSS, DREAD etc.), the likelihood continues being a complex measurement that requires either a history of vulnerabilities or an expert, who determines based on several factors (e.g., equipment, necessary knowledge, exposure of the system etc.), the likelihood that this vulnerability can be exploited. Dealing with this problem, the proposed methodology establishes a mapping between the test result and the likelihood value, obtaining it directly from the testing process. Therefore, likelihood will take three possible values, as shown in Table 2. If the test is deterministic (e.g., test: ciphered communications), that is the test PASS or FAILS, likelihood value will be 0 or 1, respectively. For example, if communications are ciphered, the test will PASS and, therefore, the likelihood of exploiting this vulnerability will be zero. However, if no ciphering is used, the test will FAIL and, therefore, the likelihood of exploiting the vulnerability will be 1. The same happens if the test is related with checking that a specific threat from the NVD is present or not in the system. More elaborated tests can also be executed that, instead of obtaining a Boolean result, a specific measure of a metric is obtained (e.g., the percentage of ciphered data in the communications or the strength of the key lengths and algorithms used). In this case, the measure can be scaled between 0 and 1 to fit the likelihood scale. It is worth noting that both types of tests can be combined or used for refining purposes, as, for example, there is no sense in measuring the percentage of ciphered data or checking the algorithm strength if no ciphering is being used.

Table 2 Mapping between the test result and likelihood

Test result	Likelihood
PASS	0
FAIL	1
Specific metric	Metric weighted between 0 and 1

In the second part of the equation, there is the impact factor, which is assigned by the security expert evaluating the system. It should be noted that not all the vulnerabilities or threats have the same type of impact on the system if they are exploited. A successful attack in the engine of an airplane could affect the safety of the people whereas an attack over the on-board screen player of the passengers would lead to a monetary impact. Therefore, the term impact as a multidimensional measure is considered. In particular, as part of the impact, the following aspects are considered: safety, financial, operational, privacy and legislation. Indeed, this approach has already been applied in other strategies such as HEAVENS, EVITA, MoRA or OWASP. Moreover, claims of D7.1 were already classified in this way to facilitate the application of the methodology by the evaluator.

Table 3 shows the safety impact metrics. The division into different criteria taken from ISO 26262-3 standard allows a clear distinction between the metrics, ranging from no injury (0 impact) to fatal injuries and survival uncertainty (1000 impact). Table 4 indicates the financial impact metrics, in this case following the classification of the BSI-Standard 100-4, from no effect over the organisation to the possibility that the company disappears. In Table 5, it is shown the impact related with the normal operation and functioning of the system, as in FMEA, where no discernible effects are evaluated with

zero impact and major disruptions of the system represent a high impact (100). Finally, Table 6 summarises the criteria of HEAVENS for the privacy and legislation impact values, from no effect to privacy violations causing significant consequences for business operation, financial aspects and for the trust and reputation of the victim. Note that financial and safety impact reach higher values than the other impact aspects [108], reflecting that usually, privacy issues will not have a direct financial or operational impact, although it might happen in some cases.

The overall impact is obtained by summing up all the metrics from the 4 impact dimensions and by mapping the result with an impact value between 0 and 10, as Table 7 indicates.

Table 3 Safety impact metrics

Safety metric	Criteria
0	No injury.
10	Light and moderate injuries.
100	Severe and life-threatening injuries (survival probable).
1000	Life-threatening injuries (survival uncertainty), fatal injuries.

Table 4 Financial impact metrics

Financial metric	Criteria
0	No discernible effect. No appreciable consequences.
10	The financial damage remains tolerable to the organization.
100	The resulting damage leads to substantial financial losses but does not threaten the existence of the organization.
1000	The financial damage threatens the existence of the organization.

Table 5 Operational impact metrics

Operational metric	Criteria
0	No discernible effect.
1	Minor/Moderate.
10	Degradation or loss of secondary function (system still operable but comfort or convenience functions don't work or work at a reduced level of performance).
	Degradation of a primary function (system still operates but at a derated performance).
100	Loss of primary function (system inoperable but does not affect its safety).
	Potential failure mode affects safe system operation with some warning or non-compliance with government regulation.

Table 6 Privacy and legislation impact metrics

Privacy and legislation metric	Criteria
0	No discernible effects in relation to violations of privacy and legislation.
	Privacy violations of a particular stakeholder, which may not lead to abuses (e.g., Impersonation of a victim to perform actions with stolen identities is not acceptable).
1	Violation of legislations without appreciable consequences for business operations and finance.
10	Privacy violations of a particular stakeholder leading to abuses and media coverage.
	Violation of legislations with potential of consequences for business operations and finance (e.g., penalties, loss of market share, media coverage).
100	Privacy violation of multiple stakeholders leading to abuses. Such a level of privacy violation may lead to extensive media coverage as well as severe consequences in terms of loss of market share, business operations, trust, reputation and finance for manufacturers and providers.
	Violation of legislations causing significant consequences for business operations and finance (e.g., huge financial penalties, loss of market share) as well as extensive media coverage.

Table 7 Overall impact level calculation

Sum of the impact aspects	Impact level	Impact value
0	No impact	0
1-6	Low	1
7-13	Low	2
14-19	Low	3
20-45	Medium	4
46-73	Medium	5
74-99	Medium	6
100-299	High	7
299-699	High	8
699-999	High	9
>=1000	Critical	10

The overall risk of the system is obtained by going from bottom to top (climbing) of the tree created during the risk identification phase, refining from the test results to each of the security properties of the system given by STRIDE, as shown in Figure 4:

- **From tests to claims**

Each claim has an associated likelihood ($Likelihood_{Claim}$) and an impact ($Impact_{Claim}$). The likelihood comes from the testing phases whereas the impact value calculation will slightly differ depending on whether the claim is directly associated with a series of tests (e.g. claim 3 of Figure 4) or whether it is associated firstly with known vulnerabilities (e.g. claim 2 of Figure 4). In any case, the risk associated with each claim will be then calculated as:

$$Risk_{Claim} = Impact_{Claim} * Likelihood_{Claim}$$

If the **claim is directly associated with the execution of *m* tests (*test-based claims*)**, the formula is refined into:

$$Risk_{Claim} = Impact_{Claim} * \frac{\sum_m Likelihood_{test}}{m}$$

Where $Impact_{Claim}$ is manually established by the evaluator, using the impact tables described before, and $Likelihood_{claim}$ is obtained as a mean over all likelihoods from the *m* associated tests. Mean was chosen instead of the max function due to the fact that different tests could measure different aspects (e.g., for confidentiality we can measure the strength of the algorithm, the strength of the key and the percentage of ciphered data).

If the **claim is associated with the existence of *n* known vulnerabilities (*vulnerability-based claims*)**, weaknesses or threats with known $Impact_{vulnerability_i}$ available in a public database the formula is refined into:

$$Risk_{Claim} = \max_{1 \le i \le n} \{ Impact_{vulnerability_i} * Likelihood_{vulnerability_i} \}$$

Max function was used here instead of the mean so as not to diminish the importance of a very serious vulnerability. $Likelihood_{vulnerability}$, similarly, as before, is obtained as a mean over all likelihoods from the *d* associated tests to each vulnerability.

$$Likelihood_{vulnerability_i} = \frac{\sum_d Likelihood_{test}}{d}$$

In both cases $Likelihood_{test}$ is the mapping between test result and likelihood that we established in Table 2.

- **From claims to components**

Once there is a risk measure between 0 and 10 per claim, it is possible to combine them to obtain the risk of a component. Therefore, if the component has *r* claims associated, the $Risk_{Component}$ is calculated as:

$$Risk_{Component} = \max_{1 \le i \le r} \{ Risk_{claim_i} \}$$

- **From components to system properties**

As already described in the risk identification phase, the system has one subtree for each STRIDE security property (confidentiality, integrity, non-repudiation, authorization, authentication, and availability). Therefore, the last step is to aggregate components into the risk associated to the correspondent security property (from the blue to the violet leaves in Figure 4). If a system property is considered in *c* components, the overall risk is calculated as:

$$Risk_{security\ property} = \max_{1 \le i \le c} \{ Risk_{component_i} * Sensitivity_{component_i} \}$$

Where the $Sensitivity_{component_i}$ was established in the risk identification phase for each component.

At the end of this process, there will exist a risk measurement for each one of the 6 STRIDE security properties in the system.

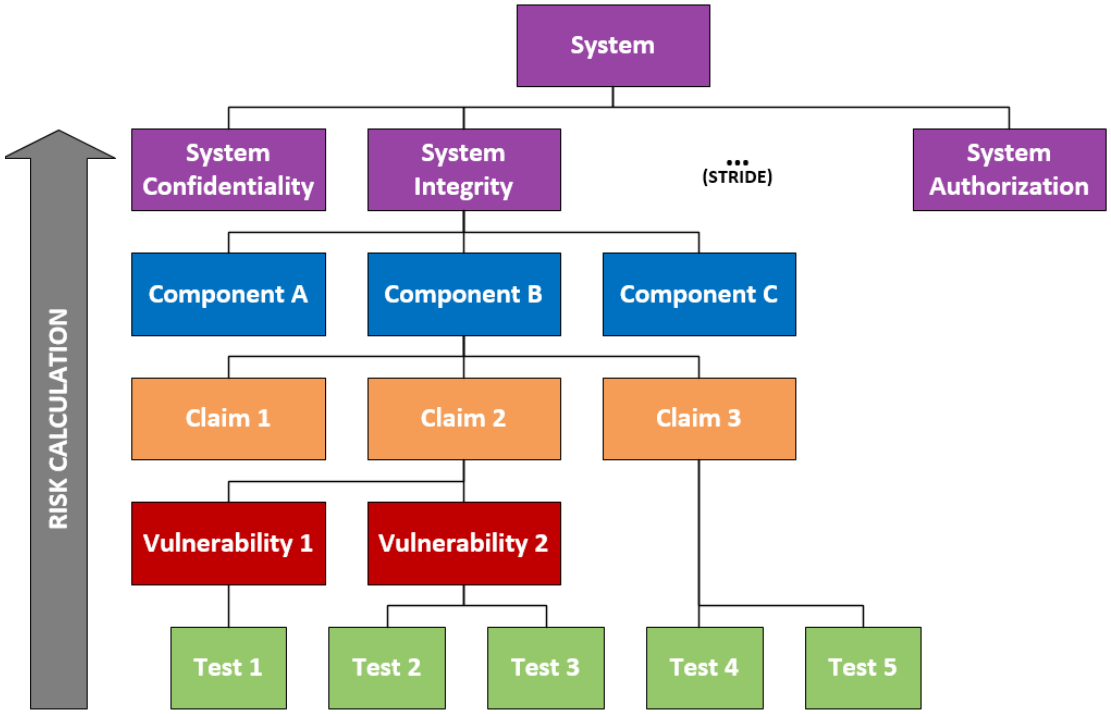

Figure 4 Risk calculation

6.5 Risk Evaluation

This phase is intended to evaluate if the estimated risk is acceptable or not for the specific context in which the system is being evaluated. Security is relative, meaning that the security properties needed in a smart home are different from the ones required for example in a vehicle. To evaluate this, the profiles established in the context phase are used.

Following the example TP defined in Figure 2, Figure 5 shows the risk evaluation process against it. Therefore, if the system obtains a confidentiality risk of 3, this value will be an acceptable risk and the system will obtain the security level B for the security property confidentiality. However, if the system obtains a confidentiality risk of 8, the risk will not be acceptable for a smart vehicle and the certificate will not be granted.

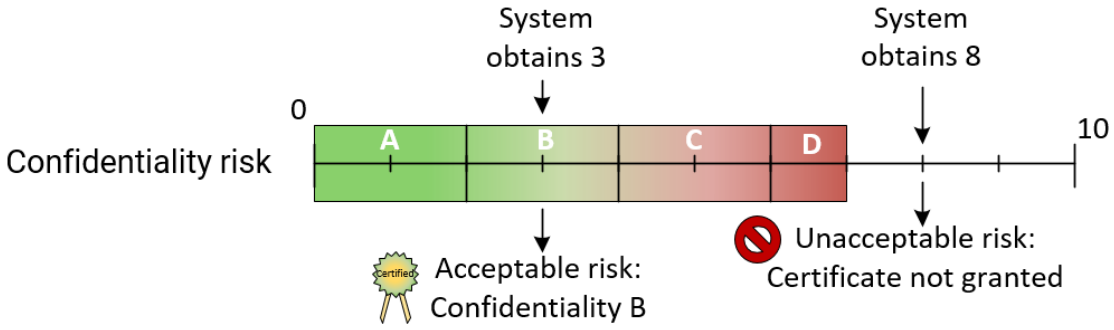

Figure 5 Example of risk evaluation against a PP

This process will be repeated for each one of the six STRIDE security properties, obtaining a security level for each of them.

6.6 Labelling

The certification process must be performed by an accredited certification laboratory [109] and, as a result, a certificate is issued to the system. In the EU, according to the recommendations of the ECSO WG1 [93], the certificate could be represented by a European Cyber Security Certificate (ECSC), which is intended to serve as an electronic booklet including different aspects of the certification process. The certificate should provide a cybersecurity label, which summarizes, in a visual way, the content of the certificate obtained as a result of the cybersecurity certification process, helping users to compare the security levels of different products, similarly to the EU Energy Label [110], which is widely used nowadays.

It is proposed the generation of a visual label integrating the information about the context in which the TOE has been certified by means of the TPs and the security level obtained for each security property. The label has been designed as a hexagonal radar diagram to support the visualization of the security dimensions (i.e. the six STRIDE security properties) and to provide a visual representation that could be understood by non-expert users. The union of the security levels and profiles creates an area that represents the overall security level of the system, meaning: **the more area, the more secure** the system is. Figure 6 shows an example of the proposed label, which follows ECSO recommendations by integrating a QR-code to deal with future updates of the label and to obtain additional information about the security evaluation process and results.

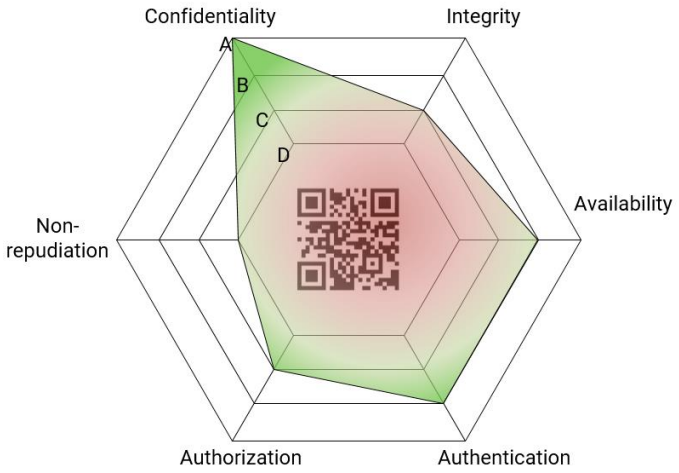

Figure 6 Example of the proposed label

6.7 Treatment

According to ISO 31000, "*Risk treatment is a risk modification process. It involves selecting and implementing one or more treatment options. Once a treatment has been implemented, it becomes a control, or it modifies existing controls*". In general, the results of the security evaluation are used only to validate or certify the security of the system, missing a very valuable information that reports the security flaws that our system has

and how they could be avoided during the system's operation phase. In this sense, we propose, as a way to address risk treatment, the integration of additional security recommendations inside the extended behavioral profile defined in D6.2 from the MUD file standard. Therefore, this activity inter-operates and makes use of the results of the previous evaluation, as the behavioral profile is intended to be generated from the security results containing both security recommendations from the manufacturer (WP6) and from the security evaluation (WP7). Possible values of the MUD file that could be obtained from the evaluation are the resistance against DoS attacks to limit the simultaneous communications or recommended cipher suite and key lengths (e.g. for ciphering or integrity protection). Further analysis of this integration will be performed in D7.3.

This profile is intended to address or mitigate security issues encountered during the evaluation process, reducing the attack surface to the allowed behaviors. Therefore, this profile could be used to enforce the recommendations provided and to monitor suspicious behaviors during the operation phase that are outside the ones reflected in the MUD file.

6.8 Monitoring and Communication

The cybersecurity certification process should not end once the initial security assessment is completed prior to market deployment. Instead, it should reflect a dynamic approach that uses complementing procedures and technologies to detect new vulnerabilities and reassess the security of the system. This is crucial, for example, to mitigate zero-day vulnerabilities, which may not be discovered during the initial security assessment. The result of the security evaluation is linked to the certification procedure, which generates a cybersecurity certificate that indicates the system's security level. In this regard, well known entities such as the NIST[111] or the European Commission (CSA [101]) emphasize that the certification process should not be halted after the initial cybersecurity examination prior to deployment but rather should be supported throughout the life cycle. Therefore, the mechanisms for maintaining an up-to-date security level throughout the lifecycle of a system should be included in the definition of a cybersecurity certification framework.

During the operation phase, the system and its components are providing the functionality for which they were manufactured. In this phase, the system should be monitored, since new security vulnerabilities can be discovered or a new patch/update can be installed, and consequently, the system’s security level can be modified. Both the changes produced by an updating process and the modifications produced by an unexpected event (e.g., the discovery of a new vulnerability) lead to a new security level, so a continuous reassessment should be done, starting a re-evaluation process if needed. In this context, the deployment of security solutions to control the behaviour of the system components and networks (e.g., the one developed within WP5) is essential to improve the reaction time by the infrastructure managers and the fast mitigation of vulnerabilities.

As mentioned before, combining the assessment results and the behavioural profile provides a highly effective tool for ensuring that the system and its component behaviour is as expected throughout the operation phase. If a deviation from normal behaviour, which could lead to an attack, is identified, a proper mitigation can be implemented to

prevent it. In sensitive contexts, where a security breach could result in the loss of human life, mechanisms such as monitoring, IDS, and mitigation solutions are critical. The communication of security information across all stakeholders becomes even more critical in this scenario. It is important to discover the existence of a vulnerability as soon as feasible and apply a patch to mitigate it in order to keep the system secure throughout its life cycle.

Finally, the cybersecurity certificate should evolve to accurately reflect the system security status. The relationship between the cybersecurity certificate lifecycle and the system's lifecycle is depicted in Figure 7, which is made up of several stages or phases. During the manufacturing phase, the cybersecurity certification process reviews and certifies the system based on security standards. Then, the system starts the operation phase, in which it is providing its intended functionality. During this phase, the behaviour of the system should be monitored to identify potential vulnerabilities which have not been detected in the initial certification process. Whatever the process to detect such vulnerabilities, this information must be shared with the manufacturer or provider of the system and/or components, according to the NIS directive [112].

To address encountered threats, the manufacturer may decide to patch or update the system (upgrading phase), which may entail changes to the system configuration. If the software changes are considerable, a new cybersecurity certification (i.e. re-certification) may be required, where system security level could be updated.

Finally, the decommissioning phase is where the component or system is decommissioned from a certain domain. Because some systems hold sensitive data (e.g., cryptographic material, private data), the method should ensure that such data is not accessible by unauthorized persons when the system is no longer operating. The system cybersecurity certificate should be properly revoked as well.

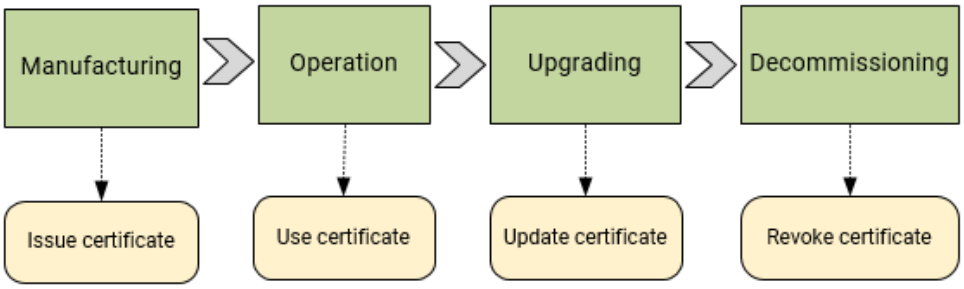

Figure 7 Certificate lifecycle

6 Conclusions

This deliverable has reported the identification of current security evaluation and certification challenges. Towards this end, diverse certification schemes, risk assessment mechanisms and testing techniques have been analysed, reflecting their strengths and weaknesses. Addressing those challenges, the deliverable has proposed a security evaluation and certification methodology, detailing each of the steps, from the identification of requirements to the creation of a visual label to represent the security level obtained. On the one hand, an innovative proposal is presented that serves as the basis towards that common European certification framework, which is the objective of the Cybersecurity Act, and, on the other hand, the methodology is designed considering the needs of the different parties involved in the certification. While dynamic mechanisms are proposed to save costs and time for the manufacturer, the consumer will be able to choose which product to buy based on the safety offered by just observing the label, in a similar way to the energy efficiency label. Moreover, the methodology has been designed in a high-level way, so that a security evaluator has the freedom to instantiate the different steps using any kind of technique or tool.

The claims defined in D7.1 have served as input for the methodology, identifying a basic set against which the system could be evaluated. Also, this deliverable will serve as input for task 7.3, focused on a particular instantiation of the methodology through concrete techniques and tools.

7 Artefacts

Table 8 shows the artefacts produced in this deliverable.

Table 8 Artefacts produced in T7.2

Name	Description
Challenges of current security certification and evaluation schemes	A list of the main challenges identified in current security certification solutions, obtained from a deep analysis of the flaws and strengths of the main security certification and evaluation schemes.
Security evaluation methodology	Definition of a set of steps and processes to evaluate the security of a system by combining risk assessment and testing. The proposed methodology is intended to be generic enough to be instantiated through different techniques and tools.

8 References

[1] MELL, Peter, et al. *CAESARS Framework Extension: An Enterprise Continuous Monitoring Technical Reference Architecture*. National Institute of Standards and Technology, 2012.

[2] *DIGITALEUROPE's views on cybersecurity certification and labelling schemes*. DIGITALEUROPE, 02-Mar-2017. [Online] Available: https://www.digitaleurope.org/resources/digitaleuropes-views-on-cybersecurity-certification-and-labelling-schemes/

[3] *WG1: Standardisation, Certification and Supply Chain Management*. ECSO. [Online] Available: https://ecs-org.eu/working-groups/wg1-standardisation-certification-and-supply-chain-management

[4] *Methods for Testing & Specification; Risk-based Security Assessment and Testing Methodologies.* ETSI, Nov-2015 [Online] Available: https://www.etsi.org/deliver/etsi_eg/203200_203299/203251/01.01.01_50/eg_203251v010101m.pdf

[5] *Methods for Testing & Specification; Risk-based Security Assessment and Testing Methodologies.* ETSI, Nov-2015 [Online] Available: https://www.etsi.org/deliver/etsi_eg/203200_203299/203251/01.01.01_50/eg_203251v010101m.pdf

[6] ARMOUR. [Online] Available: https://www.sciencedirect.com/science/article/abs/pii/S0920548918301375

[7] *Common Criteria for Information Technology Security Evaluation. Part 1: Introduction and general model.* Common Criteria, 2017.

[8] *Arrangement on the Recognition of Common Criteria Certificates In the field of Information Technology Security*. Common Criteria, 2014. [Online] Available: https://www.commoncriteriaportal.org/files/operatingprocedures/cc-recarrange.pdf

[9] C. Zhou, S. Ramacciotti, *Common Criteria: Its Limitations and Advice on Improvement*, *ISSA J.*, 2011. [Online] Available: https://www.difesa.it/SMD_/Staff/Reparti/II/CeVa/Pubblicazioni/Estere/Documents/CommonCriteria_ISSA%20Journal_0411.pdf

[10] S. P. Kaluvuri, M. Bezzi, Y. Roudier, *A Quantitative Analysis of Common Criteria Certification Practice, in Trust, Privacy, and Security in Digital Business*, vol. 8647, Cham: Springer International Publishing, 2014, pp. 132-143.

[11] F. Keblawi, D. Sullivan, *Applying the common criteria in systems engineering*, *IEEE Secur. Priv. Mag.*, vol. 4, n.º 2, pp. 50-55, 2006, doi: 10.1109/MSP.2006.35.

[12] *CyberSecurity Certification: EUCC Candidate Scheme*. ENISA. [Online] Available: https://www.enisa.europa.eu/publications/cybersecurity-certification-eucc-candidate-scheme-v1-1.1

[13] *Cloud Service Scheme*. EUCS, 22-Dec-2020, [Online] Available: https://www.enisa.europa.eu/topics/publications/eucs-cloud-service-scheme

[14] *Securing EU's Vision on 5G: Cybersecurity Certification*. EUCS, 03-Feb-2021, [Online] Available: https://www.enisa.europa.eu/news/enisa-news/securing_eu_vision_on_5g_cybersecurity_certification

[15] *Certification de sécurité de premier niveau (CSPN)*, *ANSSI*, 2008. [Online] Available: https://www.ssi.gouv.fr/administration/produits-certifies/cspn/

[16] *Certification de securite de premier niveau*. CryptoExperts [Online] Available: https://www.cryptoexperts.com/services/cspn/

[17] G. Baldini, G. Giannopoulos, A. Lazari, *Annex 8: JRC Analysis and recommendations for a European certification and labelling framework for cybersecurity in Europe*, European Commission, 2017.
[18] *UL 2900 Standards Process*. Underwriters Laboratories [Online] Available: https://industries.ul.com/cybersecurity/ul-2900-standards-process
[19] *The Commercial Product Assurance (CPA) build standard*. CESG, 2014, [Online] Available: https://www.nccgroup.trust/uk/our-services/cyber-security/compliance-and-accreditations/cpa-and-cc/
[20] *Foundation Grade explained*. National Cybersecurity Center of United Kingdom, 2017, [Online] Available: https://www.ncsc.gov.uk/articles/foundation-grade-explained
[21] *The Commercial Product Assurance (CPA) build standard*. CESG, 2014. [Online] Available: https://www.nccgroup.trust/uk/our-services/cyber-security/compliance-and-accreditations/cpa-and-cc/
[22] *STATE OF THE ART SYLLABUS*. ECSO, Jun-2017, [Online] Available: http://www.ecs-org.eu/documents/uploads/state-of-the-art-syllabus-v1.pdf
[23] *IEC 62443-4-1:2018*. IEC Webstore, 15-Jan-2018, [Online] Available: https://webstore.iec.ch/publication/33615
[24] *Security of Industrial Automation and Control Systems*. International Society of Automation (ISA), Oct-2020, [Online] Available: https://www.isasecure.org/en-US/Documents/Articles-and-Technical-Papers/ISAGCA-Security-Lifecycles-whitepaper
[25] *CNSSI 4009 Committee on National Security Systems (CNSS) Glossary,* 2015. [Online] Available: https://www.serdp-estcp.org/Tools-and-Training/Installation-Energy-and-Water/Cybersecurity/Resources-Tools-and-Publications/Resources-and-Tools-Files/CNSSI-4009-Committee-on-National-Security-Systems-CNSS-Glossary
[26] *Common Weakness Scoring System (CWSS)*. MITRE, 2014. [Online] Available: https://cwe.mitre.org/cwss/cwss_v1.0.1.html
[27] J. R. C. Nurse, S. Creese, and D. D. Roure, *Security Risk Assessment inInternet of Things Systems*, IEEE Computer Society, IT Pro, 2017.
[28] Common Weakness Enumeration (CWE), [Online] Available: https://cwe.mitre.org/
[29] OWASP Top Ten, [Online] Available: https://owasp.org/www-project-top-ten/
[30] *Common Vulnerability Score System (CVSS) v3*. IRST, 2015, [Online] Available: https://www.first.org/cvss/v3.0/cvss-v30-specification_v1.9.pdf
[31] Common Vulnerabilities and Exposures (CVE), [Online] Available: https://cve.mitre.org/
[32] National Vulnerability Database (NVD), [Online] Available: https://nvd.nist.gov/
[33] R. A. Caralli, J. F. Stevens, L. R. Young, y W. R. Wilson, *Introducing OCTAVE Allegro: Improving the Information Security Risk Assessment Process*, CERT, 2007
[34] C. J. Alberts, A. J. Dorofee, J. F. Stevens, y C. Woody, *OCTAVE-S Implementation Guide, Version 1*, 2005
[35] *DREAD scheme*. Microsoft, 2010, [Online] Available: https://docs.microsoft.com/en-us/previous-versions/msp-n-p/ff648644(v=pandp.10)#dread
[36] A. B. Garcia, R. F. Babiceanu, y R. Seker, *Trustworthiness requirements and models for aviation and aerospace systems*, 2018 Integrated Communications,

Navigation, Surveillance Conference (ICNS), Herndon, VA, 2018, pp. 1-16, doi: 10.1109/ICNSURV.2018.8384911

[37] *Threat prioritisation: DREAD is dead, baby?*. NCCgroup, 2016, [Online] Available: https://www.nccgroup.trust/uk/about-us/newsroom-and-events/blogs/2016/march/threat-prioritisation-dread-is-dead-baby/

[38] *VerAfied Methodology*. VERACODE [Online] Available: https://www.veracode.com/verified

[39] *Veracode Detailed Report.* Nextcloud, 2016, [Online] Available: https://nextcloud.com/wp-content/themes/next/assets/files/veracode_report.pdf?x53054

[40] *The Cenzic HARM (Hailstorm Application Risk Metric) Score*. Cenzic, [Online] Available: https://owasp.org/www-pdf-archive//OWASP_Cloudy_with_a_chance_of_hack_Nov_2010.pdf

[41] OWASP Application Security Verification Standard (ASVS) Project. OWASP, [Online] Available: https://owasp.org/www-project-application-security-verification-standard/

[42] R. M. R. K, *Security risk assessment of Geospatial Weather Information System (GWIS) using integrated CVSS approach*, Int. J. Adv. Comput. Sci. Appl., vol. 1, n.o 3, 2010

[43] A. R. Ruddle, H. Mira, and S. Information, *Security requirements for automotive on-board networks based on dark-side scenarios. E-safety vehicle intrusion protected applications*. EVITA project, Tech. Rep. February 2016, 2009, [Online] Available: https://www.researchgate.net/publication/46307752_Security_requirements_for_automotive_on-board_networks_based_on_dark-side_scenarios_Deliverable_D23_EVITA_E-safety_vehicle_intrusion_protected_applications

[44] L. Aljoscha and M. Islam, *HEAling Vulnerabilities to ENhance Software Security and Safety - Project Proposal HEAVENS*. 2016.

[45] ETSI TS 102 165-1 Methods and protocols; Part 1: Method and pro forma for Threat, Vulnerability, Risk Analysis (TVRA). ETSI, 2017.

[46] Macher, G., Armengaud, E., Brenner, E., Kreiner, C.: *A review of threat analysis and risk assessment methods in the automotive context*. In International Conference on Computer Safety, Reliability, and Security. pp. 130–141. Springer (2016)

[47] Palin, R., Ward, D., Habli, I., Rivett, R.: *Iso 26262 safety cases: Compliance and assurance*. In 6th IET International Conference on System Safety 2011. pp. 1–6.IET (2011)

[48] *ISO: ISO/SAE FDIS 21434 Road vehicles*. Cybersecurity engineering, [Online] Available: https://www.iso.org/standard/70918.html

[49] *SESAMO: Security and Safety Modelling.* 2015, [Online] Available: http://www.sesamo-project.eu/

[50] *SESAMO: D4.2 Integrated Design and Evaluation Methodology*. 2014, [Online] Available: http://sesamo-project.eu/content/d42-integrated-design-and-evaluation-methodology

[51] Ramaiah, B.S.M.P.S., Gokhale, A.A.*: Fmea and fault tree based software safety analysis of a railroad crossing critical system*. Global Journal of Computer Science and Technology (2011)

[52] Schmittner, C., Gruber, T., Puschner, P., Schoitsch, E.: *Security application of failure mode and effect analysis (fmea)*. In International Conference on Computer Safety, Reliability, and Security. pp. 310–325. Springer (2014)

[53] Ref: Stolte, T., Bagschik, G., Reschka, A., Maurer, M.: *Hazard analysis and risk assess-ment for an automated unmanned protective vehicle*. In 2017 IEEE Intelligent Vehicles Symposium (IV). pp. 1848–1855. IEEE (2017)

[54] Macher, G., Sporer, H., Berlach, R., Armengaud, E., Kreiner, C.: Sahara: *A security-aware hazard and risk analysis method*. In 2015 Design, Automation Test in Europe Conference Exhibition (DATE). pp. 621–624 (2015). [Online] Available: https://doi.org/10.7873/DATE.2015.0622

[55] Ishimatsu, T., Leveson, N.G., Thomas, J.P., Fleming, C.H., Katahira, M., Miyamoto,Y., Ujiie, R., Nakao, H., Hoshino, N.: *Hazard analysis of complex spacecraft usingsystems-theoretic process analysis*. Journal of Spacecraft and Rockets51(2), 509–522(2014)

[56] Young, W., Porada, R.: *System-theoretic process analysis for security (stpa-sec): Cyber security and stpa*. In 2017 STAMP Conference (2017)

[57] Raspotnig, C., Katta, V., Karpati, P., Opdahl, A.L.: *Enhancing chassis: a methodfor combining safety and security*. In 2013 International Conference on Availability,Reliability and Security. pp. 766–773. IEEE (2013)

[58] Pentti, H., Atte, H.: *Failure mode and effects analysis of software-based automationsystems*. VTT Industrial Systems, STUK-YTO-TR190, 190 (2002)

[59] Macher, G., Armengaud, E., Brenner, E., Kreiner, C.: *Threat and risk assessment methodologies in the automotive domain*. Procedia computer science 83, 1288–1294(2016)

[60] G¨oßling-Reisemann, S.: *Resilience–preparing energy systems for the unexpected*. An edited collection of authored pieces comparing, contrasting, and integrating riskand resilience with an emphasis on ways to measure resilience p. 73 (2016)

[61] Cioroaica E., Kar S.R., Sorokos I. (2022) Comparison of Safety and Security Analysis Techniques. In: Gude Prego J.J., de la Puerta J.G., García Bringas P., Quintián H., Corchado E. (eds) 14th International Conference on Computational Intelligence in Security for Information Systems and 12th International Conference on European Transnational Educational (CISIS 2021 and ICEUTE 2021). CISIS - ICEUTE 2021. Advances in Intelligent Systems and Computing, vol 1400. Springer, Cham. [Online] Available: https://doi.org/10.1007/978-3-030-87872-6_23

[62] F. Bouquet, C. Grandpierre, B. Legeard, F. Peureux, N. Vacelet, andM. Utting, *A Subset of Precise UML for Model-Based Testing,* in Proceedings of the 3rd International Workshop on Advances in Model-Based Testing - A-MOST '07. London, United Kingdom: ACM Press, 2007, pp.95–104.

[63] M. Felderer, B. Agreiter, P. Zech, and R. Breu, *A Classification for Model-Based Security Testing*, in VALID 2011, The Third International Conference on Advances in System Testing and Validation Lifecycle, 2011, pp. 109–114.

[64] D. Xu, M. Tu, M. Sanford, L. Thomas, D. Woodraska, and W. Xu, *Automated Security Test Generation with Formal Threat Models*, IEEE Transactions on Dependable and Secure Computing, vol. 9, no. 4, pp. 526–540, 2012.

[65] A. Cretin, B. Legeard, F. Peureux, and A. Vernotte, *Increasing the Resilienceof ATC systems against False Data Injection Attacks using DSLbased Testing*, in Doctoral Symposium ICRAT, 2018.

[66] W. Li, F. Le Gall, and N. Spaseski, *A Survey on Model-Based Testing Tools for Test Case Generation*, in Tools and Methods of Program Analysis, Itsykson, A. Scedrov, and V. Zakharov, Eds., vol. 779. Cham: Springer International Publishing, 2018, pp. 77–89.

[67] B. Legeard and A. Bouzy, *Smartesting CertifyIt: Model-Based Testing for Enterprise IT*, in 2013 IEEE Sixth International Conference on Software Testing, Verification and Validation. Luxembourg, Luxembourg: IEEE, 2013, pp. 391–397.

[68] *GraphWalker, an open-source model-based testomg tool*. [Online] Available: https://graphwalker.github.io/

[69] MISTA. [Online] Available: http://cs.boisestate.edu/~dxu/research/MBT.html

[70] A. Vernotte, *Research Questions for Model-Based Vulnerability Testing of Web Applications*, in IEEE International Conference on Software Testing, Verification, and Validation Workshops, 2013.

[71] J. Bozic and F. Wotawa, *Security Testing Based on Attack Patterns*, in IEEE International Conference on Software Testing, Verification, and Validation Workshops, 2014.

[72] M. Felderer and E. Fourneret, *A Systematic Classification of Security Regression Testing Approaches*, International Journal on Software Tools for Technology Transfer, vol. 17, no. 3, pp. 305–319, 2015.

[73] M. Bishop, *About Penetration Testing*, IEEE Security & Privacy Maga-zine, vol. 5, no. 6, pp. 84–87, 2007

[74] J. Bau, E. Bursztein, D. Gupta, and J. Mitchell, *State of the Art: Auto-mated Black-Box Web Application Vulnerability Testing*, in 2010 IEEESymposium on Security and Privacy. Oakland, CA, USA: IEEE, 2010,pp. 332–345

[75] ISECOM, *The Open Source Security Testing Methodology Manual (OS-STMMv3*), 2010

[76] M. Sutton, A. Greene, and P. Aminir, *Fuzzing: Brute Force VulnerabilityDiscovery*. Pearson Education, 2007, pp. 1–51.

[77] C. Chen, B. Cui, J. Ma, R. Wu, J. Guo, and W. Liu, *A Systematic Reviewof Fuzzing Techniques,* Computers & Security, vol. 75, pp. 118–137, 2018

[78] M. Schneider, J. Grossmann, I. Schieferdecker, and A. Pietschker, *OnlineModel-Based Behavioral Fuzzing*, in IEEE Sixth International Conferenceon Software Testing, Verification and Validation Workshops, 2013.

[79] P. Tsankov, M. T. Dashti, and D. Basin, *SECFUZZ: Fuzz-testing SecurityProtocols*, in Proc. of the 7th International Workshop on Automation ofSoftware Test, 2012

[80] C. Miller and Z. Peterson, *Analysis of Mutation and Generation-BasedFuzzing*, 2007

[81] W. Krenn, R. Schlick, S. Tiran, B. Aichernig, E. Jöbstl, and H. Brandl, *MoMut::UML model-based mutation testing for UML*, in2015 IEEE 8thInternational Conference on Software Testing, Verification and Validation,ICST 2015 - Proceedings, 2015.

[82] F. Duchene, *Detection of Web Vulnerabilities via Model Inferenceassisted Evolutionary Fuzzing*, Ph.D. dissertation, Grenoble University,2014. [Online] Available: https://hal.archives-ouvertes.fr/tel-01102325/document

[83] J. Bozic and F. Wotawa, *Model-based Testing - From Safety to Security*, STV Bozic, Wotawa, 2012

[84] M. Felderer, M. Büchler, M. Johns, A. D. Brucker, R. Breu, andA. Pretschner, “Chapter One - Security Testing: A Survey,” inAdvancesin Computers. Elsevier, 2015, vol. 101, pp. 1–51

[85] S. Bekrar, C. Bekrar, R. Groz, and L. Mounier, *Finding Software Vulner-abilities by Smart Fuzzing,* inFourth IEEE International Conference onSoftware Testing, Verification and Validation, 2011

[86] Yoo y M. Harman, *Regression testing minimization, selection and prioritization: a survey*, Softw. Test. Verification Reliab., vol. 22, n.o 2, pp. 67-120, mar. 2012, doi: 10.1002/stv.430

[87] E. Fourneret, F. Bouquet, Frederic Dadeau, y Stephane Debricon, *Selective Test Generation Method for Evolving Critical Systems*, in 2011 IEEE Fourth International Conference on Software Testing, Verification and Validation Workshops, Berlin, Germany, 2011, pp. 125-134, doi: 10.1109/ICSTW.2011.95

[88] L. Cseppento y Z. Micskei, *Evaluating code-based test input generator tools*, Softw. Test. Verification Reliab., vol. 27, n.o 6, p. e1627, sep. 2017, doi: 10.1002/stvr.1627

[89] B. Chess y J. West, *Secure programming with static analysis*. Gary McGraw, 2007

[90] N. Ayewah, D. Hovemeyer, J. D. Morgenthaler, J. Penix, y W. Pugh, *Using Static Analysis to Find Bugs*, IEEE Softw., vol. 25, n.o 5, pp. 22-29, sep. 2008, doi: 10.1109/MS.2008.130

[91] Mike Papadakis, Marinos Kintis, Jie Zhang, Yue Jia, Yves Le Traon, and Mark Harman. *Chapter six - mutation testing advances: An analysis and survey*. Volume 112 of Advances in Computers, pages 275 – 378. Elsevier, 2019

[92] H. Baars, R. Lassche, R. Massink, y H. Pille, «Smart grid security certification in Europe. Challenges and recommendations». ENISA, 2014

[93] *A Meta-Scheme Approach.* ECSO, Dec-2017 [Online] Available: https://www.ecs-org.eu/documents/uploads/european-cyber-security-certification-a-meta-scheme-approach.pdf

[94] S. Murdoch, M. Bond, R. J. Anderson, *How Certification Systems Fail: Lessons from the Ware Report*, *IEEE Secur. Priv. Mag.*, vol. 10, n.º 6, pp. 1-1, 2012, doi: 10.1109/MSP.2012.89

[95] S. P. Kaluvuri, M. Bezzi, Y. Roudier, *A Quantitative Analysis of Common Criteria Certification Practice*, in *Trust, Privacy, and Security in Digital Business*, vol. 8647, Cham: Springer International Publishing, 2014, pp. 132-143.

[96] AIOTI, *Report on Workshop on Security and Privacy in the Hyper-Connected World*. 2016.

[97] J. Hubner, M. Lastovka, *BOSCH Political Viewpoint. Security in IoT*. 2017.

[98] *Iot Security & Privacy Label* [Online] Available: https://iotsecurityprivacy.org/labels

[99] *CYBERSECURITY MADE IN EUROPE*. ECSO, [Online] Available: https://ecs-org.eu/initiatives/cybersecurity-made-in-europe

[100] M. Bartoletti, P. Degano, G. L. Ferrari, *Security Issues in Service Composition*, in *Formal Methods for Open Object-Based Distributed Systems*, Berlin, Heidelberg, 2006, vol. 4037, pp. 1-16, doi: 10.1007/11768869_1.

[101] Regulation (EU) 2019/881 of the European Parliament and of the Council on ENISA (the European Union Agency for Cybersecurity) and on information and communications technology cybersecurity certification and repealing Regulation (EU) No 526/2013 [Online] Available: https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32019R0881&from=EN

[102] *Methods for Testing & Specification; Risk-based Security Assesment and Testing Methodologies*. ETSI, Nov-2015. [Online] Available: https://www.etsi.org/deliver/etsi_eg/203200_203299/203251/01.01.01_50/eg_203251v010101m.pdf

[103] S. N. Matheu, J. L. Hernandez-Ramos, and A. F. Skarmeta, *Toward a Cybersecurity Certification Framework for the Internet of Things*, IEEE Security Privacy, vol. 17, no. 3, pp. 66–76, 5 2019

[104] Lorrain, E. Fourneret, F. Dadeau, and B. Legeard, *MBeeTle - un outil pour la génération de tests à-la-volée à l'aide de modèles*, in Groupement De Recherche CNRS du Génie de la Programmation et du Logiciel, 2016, [Online] Available: https://hal.archives-ouvertes.fr/hal-02472608

[105] *IoT-LAB : The Very Large Scale IoT Testbed*. FIT (Future Internet Testing Facility) [Online] Available: https://www.iot-lab.info/

[106] *HEAVENS: HEAling Vulnerabilities to ENhance Software Security and Safety* – Project Proposal, 2012.

[107] *NIST SP 800-30 Rev. 1 Guide for Conducting Risk Assessments*. NIST, Sep-2012, [Online] Available: https://csrc.nist.gov/publications/detail/sp/800-30/rev-1/final

[108] HEAling Vulnerabilities to ENhance Software Security and Safety (HEAVENS). D2: Security models (2016)

[109] *Overview of ICT certification laboratories*. ENISA, Jan-2018. [Online] Available: https://european-accreditation.org/wp-content/uploads/2018/10/document-ict-certification-laboratories.pdf

[110] *Directive 2010/30/EU on the indication by labelling and standard product information of the consumption of energy and other resources by energy-related products*. European Commission, 19-May-2010, [Online] Available: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32010L0030

[111] Katie Boeckl, Michael Fagan, William Fisher, Naomi Lefkovitz, Katerina N. Megas, Ellen Nadeau, Danna Gabel O Rourke, Ben Piccarreta, and Karen Scarfone. 25-Jun-2019. *Considerations for Managing Internet of Things (IoT) Cybersecurity and Privacy Risks*. [Online] Available: https://nvlpubs.nist.gov/nistpubs/ir/2018/NIST.IR.8228-draft.pdf

[112] *European Commission. DIRECTIVE (EU) 2016/1148 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 6 July 2016 concerning measures for a high common level of security of network and information systems across the Union*. [Online] Available: https://eur-lex.europa.eu/eli/dir/2016/1148/oj