D5.1 –Comparative Analysis of Assurance Frameworks

Security Assurance Framework for Networked Vehicular Technology

Abstract

SAFERtec proposes a flexible and efficient assurance framework for security and trustworthiness of Connected Vehicles and Vehicle-to-I (V2I) communications aiming at improving the cyber-physical security ecosystem of "connected vehicles" in Europe. The project will deliver innovative techniques, development methods and testing models for efficient assurance of security, safety and data privacy of ICT related to Connected Vehicles and V2I systems, with increased connectivity of automotive ICT systems, consumer electronics technologies and telematics, services and integration with 3rd party components and applications. The cornerstone of SAFERtec is to make assurance of security, safety and privacy aspects for Connected Vehicles, measurable, visible and controllable by stakeholders and thus enhancing confidence and trust in Connected Vehicles.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

DX.X & Title:	D5.1 Comparative Analysis of Assurance Frameworks
Work package:	WP5 Assurance Framework Evaluation
Task:	T5.1 Comparative Analysis
Due Date:	M27
Dissemination Level:	PU
Deliverable Type:	R

Authoring and review process information	
EDITOR Sammy HADDAD / OPP	**DATE** 30/03/2020
CONTRIBUTORS Sammy HADDAD / OPP Panagiotis Pantazopoulos / ICCS	**DATE** 30/03/2020
REVIEWED BY Matthieu GAY, Guillemette MASSOT / CCS Panagiotis Pantazopoulos / ICCS	**DATE** 30/03/2020
LEGAL & ETHICAL ISSUES COMMITTEE REVIEW REQUIRED?	
NO	

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

Document/Revision history

Version	Date	Partner	Description
V0.1	05/02/2019	OPP	Draft definition of all section
V0.2	14/03/2019	OPP	State of the art of existing evaluation and certification framework
V0.3	19/11/2019	OPP	Full update and extensions of comparison parameters
V0.4	24/01/2020	ICCS	Peer review report submitted
V0.5	14/02/2020	OPP	Addition of candidates for comparison. Final version submitted
V1.0	30/03/2020	OPP	Update after first internal review

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

1 Table of Contents

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

Table of Figures

List of Tables

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

Acronyms and abbreviations

Abbreviation	Description
ASN.1	Abstract Syntax Notation One
ADV	DeVelopments (CC evaluation task)
AGD	Guides (CC evaluation task)
ALC	Life-cycle (CC evaluation task)
ANSSI	Agence Nationale de la Sécurité des Systèmes d'Information
ASE	Security target Evaluation (CC evaluation task)
ATE	Tests (CC evaluation task)
CC	Common Criteria
CCRA	Common Criteria Recognition Arrangement
CERT	Computer Emergency Response Team
CSPN	Certification de Sécurité de Premier Niveau
DMZ	DeMilitarized Zone
EAL	Evaluation Assurance Level
ECU	Electronic Control Unit
GUI	Graphical User Interface
HSM	Hardware Security Module
IoT	Internet of Things
IT	Information Technology
ITS	Intelligent Transport System
IVN	In-vehicle Network
KPSI	Key Performance Security Indicators
LAN	Local Area Network
NDA	Non-disclosure Agreement
OBU	On-board Unit
OS	Operating System
PRNG	Pseudo Random Number Generator

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

SAF	Security Assurance Framework
SIEM	Security Information and Event Management
SFR	Security Functional Requirement
SOG-IS	Senior Officials Group Information Systems Security
(Open)SSL	(Open) Secure Sockets Layer
ST	Security Target
TOE	Target Of Evaluation
TSF	TOE Security Function
TSFI	TOE Security Function Interface
UNECE	United Nations Economic Commission for Europe
V-ITS-S	Vehicle-ITS-Station
WAN	Wide Area Network

Table 1: List of Abbreviations

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

Executive Summary

This deliverable presents a comparative analysis of the SAFERtec assurance framework (SAF) with the other security evaluation methods existing in the state of the art.

Comparing security evaluation frameworks is not an easy task. In fact, the state of the art itself demonstrates that none of the approaches developed during the last past 30 years has succeeded in showcasing its efficiency nor its undebatable superiority over the other approaches. No irrefutable arguments have so-far been produced on that matter.

All existing approaches are subject to different kind of criticism. They either provide very low levels of assurance (low confidence in the evaluation results) or they are considered too costly. Very few official methods have emerged and only one benefits international recognition: the Common Criteria (CC).

In order to compare different evaluation methods, the first challenge is to provide various parameters to allow for the meaningful and fair comparison of the different characteristics of each method. A second challenge is the availability of data related to real executions of those security evaluation methods in order to assess their performance. In fact, most security evaluation approaches (no matter the framework) provide confidential results, thus it is very difficult to get statistics about them (i.e. quantity and type of problem found, duration, evaluation costs, input production, etc.).

Those are the challenges we address in this deliverable to demonstrate the SAF advantages.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

1 Introduction

1.1 Purpose of the Document

The goal of this deliverable is to demonstrate SAFERtec assurance framework (SAF) efficiency and suitability to ITS domain. This will be done thanks to a comparative analysis of the SAF properties with other recognized security evaluation methods present in the state of the art.

In order to compare SAF with other evaluation framework, we will first develop a set of characteristics that will be used to evaluate comparable parameters of the different evaluation frameworks. In fact, identification of specific factors is necessary to justify why and how an evaluation framework would be better or more adapted than another one for a specific context, here C-ITS systems. There is in fact no unique and universal scale to compare different evaluation frameworks. They all have pro and cons, and different characteristics that may or may not be rated on a specific scale. Some parameters can be easy to understand and compare, like the cost to pay for an independent lab to run test (i.e. amount of euros), some other are far more complicated and cannot be compared, e.g. which approach is more likely to find vulnerability than the other: code review or black box vulnerability tests?

We propose in this document more than 20 parameters of comparison for which we try to provide easy to understand scales. Using these scales we will evaluate and compare to SAF the main references of the state of the art: FIPS (as a conformity check representative), CSPN and regular vulnerability tests services, CC and CARSEM (as assurance framework representatives), ETSI ISI 003 (as security metrics evaluation process representative) and finally ISO 21434 (as general best practice evaluation scheme representative).

1.2 Intended readership

Besides the project reviewers, this deliverable is addressed to any interested reader (*i.e.* Public dissemination level).

1.3 Inputs from other projects

This study reuses some of the comparison parameters identified in the ISE IRT system project (CARSEM: A Cooperative Autonomous Road-vehicles Security Evaluation, 17-21 September 2018) and extend them into a more exhaustive and structured set.

1.4 Relationship with other SAFERtec deliverables

This deliverable discusses the SAF proposed in the D3.1. Some of the parameters and evaluation estimations will be discussed in deliverable 5.2 and 5.3 which will respectively demonstrate the framework efficiency thanks to simulation approaches and discuss composite evaluation (to be compared to existing processes).

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

2 Evaluation framework comparison parameters

Before comparing different evaluation schemes and methodologies, we start by recalling what was already mentioned in the deliverable D3.1, i.e. the main and most important aspects that differentiate existing security evaluations methods.

When looking closely to the existing IT security evaluation methods, we see that they all address the following three dimensions:

- What must be evaluated?
  - Which product? Which version of the product? Which function of the product? In which environment? For which threat? Etc.
- Which evaluation activities?
  - Evaluating the development, evaluating the product architecture, testing the external/internal interfaces (i.e. black, white, grey box), code review, user and administration guides review, operational metrics, etc.
- Who is competent and who is in charge of what:
  - Who is the evaluation authority in charge of defining and managing the evaluation activities to guarantee the overall evaluations expectations?
  - Who will pay and be the sponsor of the evaluation?
  - Who has the expertise and required test environment?
  - What data does the developer have and what information must he provide for the evaluation of its product?
  - What is the end user's point of view?

The above three dimensions correspond to what is called by the CC (ISO/IEC, 2009) and most of evaluation experts:

- The Security Target (ST)
- The assurance components
- The evaluation scheme

All IT security evaluation schemes have their own interpretation of what is important for these three dimensions and how to obtain them. It is important to understand that there is no universal solution for the problem of IT security evaluation and all known solutions are criticized. In fact, they all have different advantages and drawbacks.

Security evaluation is a difficult problem and will probably remain so for a long time. This is due to the fact that IT systems are complex and they evolve rapidly. Whether it is feasible or not to have full formal proofs of systems' security, the current state of the art for IT technologies demonstrates that the effort is not worthy, because it is too complicated (if ever possible) and too costly for software that has an average lifetime of month see weeks or days. On the other hand, vulnerability tests are never sufficient to guarantee the absence of a vulnerability since their exhaustiveness can never be demonstrated. So, security evaluation is always stuck there in between doing nothing and having low to no confidence and over costly approaches. The challenge is to provide enough evidence of the absence of vulnerabilities when no existing or affordable framework can fully demonstrate it.

ITS systems are directly concerned by this observation. These systems are relatively new, so they do not benefit from years of real security experience, they are also complex (system of systems, large

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

applications, etc.) but they need a high level of confidence due to the high risks they are facing (possible car crashes and deaths).

In this section, we will try to identify a scale and a set of parameters that can help to compare different evaluation frameworks and analyse their pros and cons in terms of the achievable (final) assurance level, required investments for the different actors involved in the evaluation and finally the adaptation to ITS characteristics.

Most of the following parameters won't be ratio scales (scales with fully comparable values) but rather ordinal (scale on which values can be sorted) or nominal scales (where elements are only differentiated by their names (Stevens, 7 June 1946)). The different scales will be defined empirically and can be argued, but never the less they all identify parameters that have an impact on the evaluation method and its adoption.

To our knowledge no such (ratio) scale has ever been produced, frameworks are generally only debated on unformal grounds. The work we present here does not claim to solve that problem but only provides one comprehensive (yet debatable) set of scales that provide common grounds for discussion.

2.1 Comparing and evaluating assurance

The first set of parameters we present are parameters that help us compare the final level of confidence obtained on the fact that the target product or system provides its expected security requirements.

Those parameters depend on the set of evaluation actions performed and how each of these actions demonstrates the security conformity of the target.

2.1.1 Evaluation tasks

We present in this section the set of evaluation activities that are regularly used in evaluation processes proposed in the state of the art. Many approaches have been defined: (Measuring Cyber Security and Information Assurance: a State-of-the-Art Report, 2009) (ANSSI, 2014) (Clark, 2005) (ETSI, 2018) (Freiling, 2008) (ISO/IEC, 2009) (Jaquith, 2007) (Jianxin Li, 2012) (Measuring Cyber Security and Information Assurance: a State-of-the-Art Report, 2009) (NIST, 2007), etc. Very few are really used. For the sake of clarity and concision we won't address them all ((Measuring Cyber Security and Information Assurance: a State-of-the-Art Report, 2009) presents several hundreds of those references) especially knowing that it is difficult if not meaningless to compare little to unused technics with mature ones.

Actually one very important thing to note is that most of those tasks are actually falling under the scope of the evaluation tasks described by the CC (ISO/IEC, 2008) and (ISO/IEC, 2008). Those descriptions may not be the only ones or perfectly accurate to describe a specific methodology or

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

evaluation refinement. However, they are generic enough to be taken as a reference. For example, the Development class (ADV) defines several evaluation families including ADV_IMP which covers possible code review. Thus, CC defines the code reviewing activity as follow in the ADV class which can be used to define such approaches as (NIST, 2007):

> Class ADV: Development
> […]
> When documenting the security functionality of a TOE, there are two properties that need to be demonstrated. The first property is that the security functionality works correctly; that is, it performs as specified. The second property, and one that is arguably harder to demonstrate, is that the TOE cannot be used in a way such that the security functionality can be corrupted or bypassed.
> […]
>
> (ISO/IEC, 2008)***, page 121***
>
> […] when performing source code analysis covered in the ADV_IMP subactivity
> […] the following checklist can additionally be used in searching for problem areas:
> a) In the language definition, phrases such as "the effect of this construct is undefined" and terms such as "implementation dependent" or "erroneous" may indicate ill-defined areas.
> b) Aliasing (allowing the same piece of memory to be referenced in different ways) is a common source of ambiguity problems.
> c) Exception handling (e.g. what happens after memory exhaustion or stack overflow) is often poorly defined.
> Most languages in common use, however well designed, will have some problematic constructs. If the implementation language is mostly well defined, but some problematic constructs exist, then an inconclusive verdict should be assigned, pending examination of the source code.
> […]
>
> (ISO/IEC, 2008) ***page 258***

In the same way the CC describe vulnerability tests activities as follows that could also be used to describe some approaches such as (ANSSI, 2014) (Ari Takanen, 2018) (Clark, 2005) etc:

> Vulnerability analysis (AVA_VAN)
> […]
> Vulnerability analysis is an assessment to determine whether potential vulnerabilities identified, during the evaluation of the development and anticipated operation of the TOE or by other methods (e.g. by flaw hypotheses or quantitative or statistical analysis of the security behaviour of the underlying security mechanisms), could allow attackers to violate the SFRs.
> (ISO/IEC, 2008)***, page 184***
>
> The purpose of the vulnerability assessment activity is to determine the exploitability of flaws or weaknesses in the TOE in the operational environment. This determination is based upon analysis of the evaluation evidence and a search of publicly available material by the evaluator and is supported by evaluator penetration testing.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

> […]
> evaluator's independent vulnerability analysis should consider generic potential vulnerabilities under each of the following headings:
> a) generic potential vulnerabilities relevant for the type of TOE being evaluated, as may be supplied by the evaluation authority;
> b) bypassing;
> c) tampering;
> d) direct attacks;
> e) monitoring;
> f) misuse.
> […]
> (ISO/IEC, 2008) ***page 311-321***

In the same way attack trees methodology are also falling under either functional tests or vulnerability tests defined by the CC, evaluation of developer best practices under the life cycle evaluation tasks, etc.

Thus, the following tables list all known evaluation activities in the state of the art (mostly presented in (Measuring Cyber Security and Information Assurance: a State-of-the-Art Report, 2009)). It identifies all the activities being described by the CC. It extracts from the CC short descriptions of those activities (mildly adapted here). It also identifies the very few that are not linked to any task defined by the CC.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

Security target evaluation			Evaluation of the content of the document specifing the evaluation context and objectives. In fact any security evaluation has to define either explicitely or not, what is the target (system/product) to be evaluated, which security function to evaluate, in which environment.
Life-cycle	Development quality process evaluation	Configuration management and scope	This evaluation task covers the control in the processes of refinement and modification of the TOE and the related information. Configuration management systems are put in place to ensure the integrity of the portions of the TOE that they control, by providing a method of tracking any changes, and by ensuring that all changes are authorised. This provides assurance by ensuring that the developments are well controlled and thus the product quality correctly enforced.
		Lyfe-cycle definition	Using a life-cycle model that has been approved by a group of experts (e.g. academic experts, standards bodies) improves the chances that the development and maintenance models will contribute to the TOE meeting its security requirements as identified by the developer. The use of a life-cycle model including some quantitative valuation adds further assurance in the overall quality of the TOE development process.
		Tools, techniques and standards	Tools and techniques is an aspect of selecting tools that are used to develop, analyse and implement the TOE. It includes requirements to prevent illdefined, inconsistent or incorrect development tools from being used to develop the TOE. This includes, but is not limited to, programming languages, documentation, implementation standards, and other parts of the TOE such as supporting runtime libraries.
	Development security	Physical	Development security is concerned with physical, procedural, personnel, and other security measures that may be used in the development environment to protect the TOE and its parts. It includes the
		Logical	
	Delivery		Evaluation of the secure transfer of the finished TOE from the development environment into the responsibility of the user. In order to further garantee that the product has not been maliciously tampared.
	Flaw remediation		Flaw remediation requires that discovered security flaws be tracked and corrected by the developer.
	Description provided by the CC		
	Description not provided by the CC		

Table 2 Evaluation activities (I)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

Product specification and conception	Fonctional specification		This family provides assurance directly by allowing the evaluator to understand how the implemented security functions meet the identified security requirements the product should fulfill.
	Security architecture		Evaluation of the security architecture description that describes the self-protection, domain separation, non-bypassability principles, including a description of how these principles are supported by the parts of the TOE that are used for TSF initialisation.
	Implementation representation		The internal workings of the TOE may be better understood when the TOE design is analysed with corresponding portions of the implementation representation. Source code or hardware diagrams and/or IC hardware design language code or layout data that are used to build the actual hardware are examples of parts of an implementation representation.
	Code/TOE structure complexity		A TSF whose internals are well-structured is easier to implement and less likely to contain flaws that could lead to vulnerabilities; it is also easier to maintain without the introduction of flaws.
Functionnal tests	Offline/laboratoy	Evaluation of developers' tests	This family contributes to providing assurance that the likelihood of undiscovered flaws is relatively small. It evaluates that either the developper or an independant tester has validated all the security
		Independant tests	
Vulnerability analysis	Search of public domain sources to identify potential vulnerabilities		Vulnerability assessment class addresses the possibility of exploitable vulnerabilities introduced in the development or the operation of the TOE.
	Penetration testing,		
Guidance documents review			Activities consisting of evaluating for all users that all relevant aspects for the secure handling of the TOE are presented.
Operationnal	Evaluation of developers' tests		Activities consisting of evaluating developper tests or making indepent tests in operationnal context. In fact, functional tests done off line, either by the developper or by an independant evaluator does not fuly garantee that the same results would be obtained in the real operational evironnement which might be different from the test environment. Yet security is an operationnal property. The impact of such thing as having good security product poorly configured or users by passing secuirity functions cannot be evaluated off line before knowing them. That's why this evaluation activity provides an additionnal level of assurance in the real operationnal system.
	Independant tests		
	Metrics		
	Configuration audit		
	Description provided by the CC		
	Description not provided by the CC		

Table 3 Evaluation activities (II)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

Thus, we can see that CC both identifies and enforces the use of an exhaustive list of cyber-security assurance activities. Only operational tests are outside the scope of CC evaluation. Otherwise all the identified approaches in section 3, can be associated to one or sometimes several CC evaluation tasks: conformity tests correspond to a specific case of either developer's or independent functional tests, vulnerability tests are a sub part of the vulnerability analysis (usually done in a less structured way when performed outside the scope of CC evaluations), more general assurance approaches have not so far introduced new activities and/or other more specific activities such as formal proofs, use of code analysis tools, attack surface measurements, attack tree models, etc; those are either covered by the Security architecture, the Implementation representation or Code/TOE structure complexity evaluation tasks.

In the same way, more developer-centric validation approaches are also covered by Life-cycle evaluation activities (self-assessment, quality management, good practises enforcement, etc.).

The CC evaluation tasks definition cannot always be directly applied to other evaluation approaches. In the first place all evaluation activities are dependant to each other's, introducing interdependent definitions. Also, all evaluation tasks make references to the ST or the SFRs within it which is not something you have in other approaches. However, the concepts behind are the same.

2.1.2 Assurance evaluation and comparison parameters

First of all, we identify the different elements and parameters that constitute an assurance level.

Assurance is the constitution of a set of elements of proof that a product or a system fulfils its requirements. This holds for security but it could also be true for any other type of requirement (quality, safety, etc.). Thus, a level of assurance must represent the quantity of proofs provided to demonstrate that conformity. If it is understandable that adding a new proof to an already existing set of proofs, should in most cases increase the newly obtained level of assurance, and thus make comparable the two assurance levels: the first one being lower than the second since it provides strictly less elements of proof. It is unclear, if not impossible, to compare two completely different sets of proofs, and thus define which one provides more assurance.

In fact, the first thing to understand when trying to compare the final assurance level obtained when comparing two different approaches is that the only formal rule that applies is:

The assurance provided by an evaluation process EP_1 is greater or equal to the one provided by an evaluation process EP_2 if and only if EP_2 is a subset of EP_1.
$A(EP_1) \geq A(EP_2)$ iff $EP_2 \subseteq EP_1$

In fact, the state of the art has never been able to provide such formal proofs as for instance, code-review provides more assurance than black box vulnerability tests (i.e. tests without knowledge of the implementation). Whoever has practiced those two activities knows that they both help finding vulnerabilities, but no large-scale study has (statistically) demonstrated the effectiveness of the one over the other (experimenting with a large number of products and different experts applying the two methodologies). This approach would be very interesting but the cost to do it is irrelevant for any national, academic or private industry, just to compare 2 approaches among many others.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

An additional difficulty if such a study would be carried-out is the confidentiality issues related to most security analyses. Most private companies do not wish to have their products tested if they know that the results can be shared or publicly compared. If not impossible to overcome, this is a problem to consider. For example, if all CC schemes which run hundreds of evaluations every year, wanted to gather statistical data, they simply could not because evaluation reports are confidential. Another big issue is that each product is evaluated only once so such a study would not contain comparison of different comparable evaluations on the same target.

Thus, many entities worldwide (either private companies or public authorities) gather different *empirical knowledge* on the performance of each security evaluation method. But none can provide a complete statistical study with publicly demonstrated results for all the aforementioned reasons.

That's why in this deliverable, we neither aim nor can provide a (full) statically-justified comparison of the different security evaluation methods. We will identify as clearly as possible the different tractable points of comparison and then provide an empirical feedback based on the SAFERtec's consortium experience which includes recognized academic and industrial experts. Our comparative analysis is not expected to reach consensus among interested researchers or avoid critique; however, the presented work in this deliverable seeks to provide a fair, meaningful (and as accurate as possible) comparison.

To limit as much as possible, possible biases, we will try to identify quantifiable parameters to be used for comparison. For different approaches a different set of those parameters (attributes) might be quantifiable. They are provided to 'define' a reference set and drive the involved comparisons. Allowing us not to make pure allegations.

2.1.2.1 Quantifiable

The first parameter that we identify here is the property of an evaluated task's result to be quantifiable or not (i.e. qualitative versus quantitative observations).

The first intuitive goal when looking for an evaluation method is to seek for quantifiable results. The ultimate goal is to find a universal scale enabling to rate security on an integer or fraction value scale, providing comparable data for any product. This is typically the goal of security metrics approaches, e.g.: attack surface (captured by the number of open ports), code complexity (captured by the number of functions per class, depth of call, etc.), numbers of audited elements during tests or execution (captured by the number of logs generated, number of different logs, etc.), complexity of the interfaces (captured by the number of parameters to be configured, number of different windows, etc.).

Such evaluation figures would then be (supposedly) easier to calculate and thus security would become easy to evaluate. But the thing is that such figures do not usually represent a real ratio scale (wikipedia) with 1 being always twice less than 2 in terms of security. For example, a PC with two open ports with well configured VPN and SSH not subject to known vulnerabilities and using recognized

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

strong cipher suites are more secure than a PC with only HTTP port 80 open over an old vulnerable version of OS and an internet navigator.

Security is not something to be evaluated; but rather a property to be assessed.

Actually, mature evaluation methods tend to provide quantitative results, being mainly binary: yes, vulnerabilities where found or no, no vulnerability where found.

In the case of assurance methods this is further completed with a value quantifying the assurance (e.g. definition of EAL 1 to 7 in the CC) coming with the full evaluation report. And in case of simple penetration testing the result is just associated to the evaluation report.

In one's mind, a perfect evaluation method would return a result on a ratio scale, but in reality, results are nominal. E.g. the security assesses by an evaluation of firewall with 156 rules, 5 open ports, with no problem found after two days of fuzzing is simply different from the security assessed for a second firewall with 56 rules, 2 open ports and no problem found after one day of fuzzing. Even in cases such as the example provides were evaluations parameters are on ratio scale, the final comparison of the assessed security is not possible. It's just different. Likewise, a VPN tested during 5 days by an expert and a second one tested during 6 months by a layman each one finding no vulnerability, does not indicate which one is "more" secure than the other.

So, for this parameter we propose the following ordinal scale to compare it:

1 - Qualitative - nominal
2 - Qualitative - ordinal
3 - Quantitative

Quantitative being a security scale which so far has never been provided. All observed evaluation parameters in the state of the art are for most of them nominal (just different named value, e.g.: windows ≠ linux, 2 open ports ≠ 3 open ports, ECDSA ≠ RSA) and for very few of them ordinal (comparable values, a password of length 20 characters not included in any dictionary is harder to guess than a 10 character long one).

2.1.2.2 Reproducibility

The second parameter we propose to compare is the level of reproducibility of the evaluation.

On one hand there are random tests, not necessarily meaning fuzzing because to some extent fuzzing is reproducible (i.e. parameters to be fuzzed, duration of the tests, etc.) and on the other hand there are conformity checks that can be repeated (almost) in the exact way and indefinitely depending on how precisely the tests are defined and the extent to which the environment configuration is easily enforced.

For this parameter we propose the following scale:

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

The level of reproducibility of a set of evaluation task (assurance requirement) for a specific set of security requirements, i.e. for comparable security targets, depends on how much (which percentage) of the tests and verification done by the evaluator can be directly reused without modification

0- not reproducible
1- 20% reproducible
2- 40% reproducible
3- 60% reproducible
4- 80% reproducible
5- fully reproducible

2.1.2.3 Comparability

The third parameter we define is to some extent an overlap of the first two.

The question here is, whether the obtained (evaluation) results can be compared when using the same method twice on two different products or systems; in other words, the question for TOEs with different security requirements is whether the results are comparable or not. For example, are two vulnerability test campaigns of the same duration carried-out by the same expert on a firewall or on an operating system, both finding no vulnerabilities, comparable? In the same way, are two campaigns of cryptographic conformity tests run for the first one on a crypto library and the second one a Single Sign On (SSO) system, comparable if the obtained results are the same? Or finally, are two vulnerability tests run during the same amount of days on the same product by two different evaluators without any knowledge of their expertise comparable?

There are actually many parameters that could influence comparability. Here we identify the importance of these parameters as typically the final level of achievable assurance is not high when it cannot be compared to anything else or any other evaluation results.

The scale we propose for the considered parameter is the following one:

0 - No comparison is possible between different evaluations
1 - Provides elements of proof to be partially comparable for similar products (e.g. two firewalls, two OSs, two cryptographic modules)
2 - Fully comparable elements of proof for similar products (in terms of functionalities, e.g. VPNs, firewalls, etc.)

2.1.2.4 Efforts needed to interpret evaluation results

In most cases when evaluating a product or a system, the observed data or behaviour during tests or measurements are not directly interpretable. For instance, when looking for buffer overflows, or misinterpreted special character, the observed behaviour is usually an error of the product or system and not directly the opening of a shell with root access (segmentation fault, pop up with an error message, etc.).

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

Also, in many cases an observed potential vulnerability needs to be discussed regarding the product (supposed) operational environment or the evaluation hypothesis. Indeed, sometimes you can find sensitive data sent in clear (text) but only on a dedicated (not connected) physical link. Sometimes a password appears in an http request, but it is only sent over VPN, etc.

Another challenge is the type of data to be reviewed in order to interpret tests results. Some testing tools will simply produce a report clearly identifying known vulnerabilities, or some conformity tests will just return 'yes' or 'no' results, while some other tests will force the evaluator to review binaries, ASN.1 data, captured packets of unknown protocols and/or really large log files. And depending on the evaluator knowledge, they might find data that helps to identify vulnerable libraries, indication of weak implementation choices, lack of data sanitization, etc.

Thus, tests during evaluations can produce from really easy-to-read to fully straight-forward results to interpret while others provide data that take a lot of time and expertise to understand.

The scale we propose for that parameter is the following one:

0 - The observed results are not exploitable (too small data set, meaningless data set, too complex interpretation, etc.)
1 - Subjective results that can be interpreted in different ways by different experts (with possible lack of consensus)
2 - None subjective results which require security experts to be interpreted
3 - Results need no interpretation

2.1.2.5 Exhaustiveness

One difficulty in testing security is usually the large number of possible parameters or code execution of the product. Today's IT products and systems are composed of millions of lines of code each increasing exponentially the number of possible executions of the target. It is rarely, if ever possible for the human brain to known all the possible executions. Thus, when testing security only a small number of executions will be tested compared to the real number of those. That's why for instance, approaches such as fuzzing exist. They use statistics to make sure that a representative (large-enough) number of executions has been tested. Otherwise analysis of tests coverage and depth is difficult and maybe not fully exhaustive (since often it is done automatically and looking for only limited outputs).

There is no known coverage scale for a specific set of tests. A scale that would help to identify if all important parameters (or at least security relevant) have been tested sufficiently. Only in specific cases we can ensure that proofs are equivalent to a full coverage (of all potential executions), e.g. formal proofs. Similarly, no tests at all suggest that no coverage is achieved. But in between it is difficult to estimate the exhaustiveness of the ran tests.

The only assurance formula on that matter is the following (as for the general assurance formula proposed in section 2.1.2).

If one partial set of tests (PST) PST1 is a superset of another partial set of tests PST2 then the assurance provided by PST1 is greater or equal to PST2.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

The only point that can be studied is the coverage of the interface and input parameters that have been tested at least once. The CC for instance differentiates two types of coverage: coverage of the interfaces and depth of the tests for those interfaces. The first one studies the coverage of the accessible interfaces and the different functions and parameters it externally exposes while the second one explores the coverage of the internal structure (from modules coverage for the lowest assurance level up to code proofs for the highest).

But this approach only establishes that the target has been tested against its functional specification. It does not cover vulnerability tests, which is even harder to assess. The exhaustiveness of the evaluation is achieved through an examination of the developers' evidence of correspondence. Thus, this analysis activity is most of the time not possible in other evaluation contexts than CC evaluations. No other approach requires such evidences. It is to be noted that even though CC require this type of study for functional tests and not vulnerability tests this implies that evaluations demonstrate that the security functions have all been tested up to a certain level.

The scale we propose for that parameter is the following one:

0- Null - No test is done
1- Partial with no coverage evidences
2- Partial with demonstration of interface coverage
3- Exhaustive - formal proof that all executions have been tested

2.1.3 Evaluation Scheme

2.1.3.1 Level of recognition

Since assurance is all about trust and especially by peers (other developers, security experts or national governance) it is clear that the level of recognition provides an important parameter when comparing different assurance approaches.

First of all, it provides a certain assurance impact. The more the approach is recognized the more confidence is built over its results and therefore provides more assurance.

The second impact of the level of recognition is industrial, since the same evaluation results can be provided to everyone recognising the framework, which is something very important in the ITS context. If a result is recognized only 'locally' (i.e. of limited recognition: Europe, North America, China) then the ITS actors will have to go through possibly many local evaluation frameworks to demonstrate the security of their products. A world-wide approach would obviously be much more efficient than an instance having to fulfil local cyber-security evaluation requirements.

The scale we propose for that parameter is the following one:

0 – No one recognizes the evaluation scheme besides the one who defined it
1 - Existence of a community (public, academic or industrial) de facto adopting it by using it
2 - Officially recognized by one country

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

3 - Officially recognized by several countries
4 - Universally recognized

2.1.3.2 Level of maturity

Something that is often linked to the previous parameter (level of recognition) is the level of maturity of the evaluation framework. Again, trust is gained by demonstrating (to the community) that the evaluation framework efficiently works. Usually the efficiency is demonstrated by the number of problems detected during evaluations, enforcing updates and security improvement of the tested security targets. This is the case for a framework that has been sufficiently run (usually more than hundred times) on several different targets. This implies that the framework has reached a certain level of maturity.

The global level of maturity of a complete framework that includes several evaluation activities depends of two things. The maturity of the full framework (as a whole) on one hand, and the maturity of each included evaluation tasks on the other hand. Considering SAF as an example, the framework uses evaluation activities that have been extensively run in the context of CC evaluations, while the full framework itself is brand new and has never been used yet. That's why we think it's important to decouple those two dimensions.

The scales we propose for the maturity parameter are the two following ones:

Scheme
0 - Never used
1 - Used on a limited set of products (<10-15) by limited set of evaluation labs (<5)
2 - Used for several years (>2) on a large set of products (>50) by several evaluation labs (>5)
3 - Used for decades (>10 years) on a large variety of products (>1000) by many evaluation labs (>30)
Assurance activities
0 - Never used
1 - Used on a limited set of products (<10-15) by limited set of evaluation labs (<5)
2 - Used for several years (>2) on a large set of products (>50) by several evaluation labs (>5)
3 - Used for decades (>10 years) on a large variety of products (>1000) by many evaluation labs (>30)

2.1.3.3 Assurance continuity

An important aspect for the developer going through an evaluation process is the assurance continuity; it is the principle used to validate an updated version of a product that has already been evaluated.

Existing evaluation frameworks most of the time require a full re-evaluation of a product if it has been updated. This is due to the fact that any changes in the code, no matter how small it is in terms of line of codes or configuration parameters changed, can introduce vulnerabilities. Sometimes they are easy to spot and sometimes not; even knowing the modification and having justifications for the updates,

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

it can provide a false sense that there is no security impact. A full evaluation may be needed to notice that.

Examples of minor and justified changes that should have had no impact on security are numerous. One old but very representative one is the modification of PRNG code of the OpenSSL library in 2006 (Project, 2006), that consisted of commenting only two lines of code because they tended to generate warnings, which in the end reduced the entropy of the PRNG to only 32,767 choices for the seeds and thus rending any key generation by OpenSSL vulnerable to brute force attacks. During the following month before the implied vulnerability was discovered all internet communications that were encrypted thanks to OpenSSL generated keys could be subject to a brute force attack. In the same way, it is regularly observed by evaluation laboratory during evaluation processes that developers send different versions of their product claiming that there is no impact on the security, while regularly it is assessed that there is.

This is the reason why an updated version of a product can never totally benefit from its ancestor evaluation. At least an impact assessment has to be done, but as aforementioned developers' impact analysis tends to be biased, and only very detailed and well justified ones can be sufficient. Nevertheless, in the worst case scenario (re-do the whole evaluation) many of the efforts previously done can be largely reused to speed up the new evaluation, e.g. some vulnerability tests can be directly replayed without any adaptation, functional test or conformant tests as well, developers inputs are usually similar and easy to review looking only for differences, etc.. In the case where the evaluator is the one that performed the initial evaluation, the new evaluation speed-up can rich 50% or more.

Thus, a trade-off has to be defined and estimated. There are different approaches for that, and assurance continuity can be efficient. But it is very important and especially in the ITS context to be able to evaluate product updates as quickly and efficiently as possible.

The second aspect of assurance continuity relates to the duration of the validity of the evaluation results. In fact, even if a product does not change, the state of the art does. It's obvious that a product evaluated 20 years ago would not provide guarantees regarding current state of the art. So, there is a real concern regarding the validity period of an evaluation result, no matter if the evaluation framework explicitly identifies it or not.

The scales we propose for that parameter are the following ones:

Re-evaluation cost

0- Evaluation costs reduction for new evaluation no matter the level of changes 0%
1- Evaluation costs reduction for new evaluation of less than 25% code changes > 25%
2- Evaluation costs reduction for new evaluation of less than 25% code changes > 50%
4- Evaluation costs reduction for new evaluation of less than 25% code changes > 75%
5- Evaluation costs reduction for new evaluation of less than 25% code changes > 90 %

Evaluation result expiration

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

0- less than 1 month
1- less than 6 months
2- less than 1 year
3- less than 5 years
4- less than 10 years
5- no limitation

2.1.3.4 Evaluator expertise validation

It is easy to understand that in every case where the evaluation methodology is not fully exhaustive or tests are (strictly) predefined, the evaluator has to take decisions regarding which tests will be run. In that case, the expertise of the evaluator directly impacts the evaluation result and thus that the assurance provided in the end directly depends on the evaluator's competences. So, if there are no evidences of these competences the final assurance level is decreased.

Knowledge of the evaluator level of expertise can be provided only by cyber-security peers. The way it is evaluated is a real challenge clearly out of scope of this document. What we suggest here is that the validation of the evaluator's expertise provides more assurance than none. The validation can be expressed in different ways. It can be validated through a simple audit report delivered either by a private or public entity. It can be an official certification going through formalized certification process (e.g. ISO 17025 as for CC ITSEF or 17065 certified entities). We won't debate or argue the different pros and cons of every possible approached. We will use the generic term of notified bodies hereafter and validate the fact that notified bodies provide more assurance than none-notified ones.

Also, as for assurance continuity, the moment when the latest validation of the evaluator's expertise took place, is an important factor.

The scale we propose for that parameter is the following one:

0- No validation of the evaluator expertise by peers
1- Evaluator expertise validated once by peers
2- Evaluator expertise validated periodically (less than 2 years)
3- Quality and cyber-security expertise evaluated by peers (less than 2 years)

2.1.3.5 Independency of the actors

In view of personal (and/or financial interests) that might influence the evaluation process, the independency of the evaluator (mainly with respect to common financial interests) is an important factor. The more the independency of the evaluator is demonstrated the best it is in terms of gained assurance.

The scale we propose for that parameter is the following one:

0- Tests run by the developers themselves

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

1- Third party tests with no demonstration of independency of the evaluator regarding sponsor or certificatory
2- Third party tests with demonstrated financial independency

2.1.3.6 Evaluation review

For the same reasons as for the evaluator expertise validation, reviewing each evaluation report provides more confidence in its content and thus in the evaluation itself especially if the review is made by an independent peer. It has a first advantage that it forces the evaluator to provide a minimum quality for its work. The second advantage being that the reviewer can provide an extra expertise and enforce additional work.

Also, it helps to harmonize evaluation quality between different evaluators.

The scale we propose for that parameter is the following one:

0. No review of the evaluation report
1. Internal review of the evaluation report
2. Third party review of the evaluation report
3. Third party review of the evaluation report and harmonization of the reviews by several reviewing entities

2.1.3.7 Difficulty to gather expected element of proof

An evaluation scheme can require many different kinds of inputs (elements of proof) to be provided by the developer and not only the target. Some may be easily provided, e.g. the target, while others are much more complex either because of confidentiality reasons or expertise reasons.

Requiring formal proofs from the developer or a detailed internal description of the target may imply a large effort from them if they don't have the relevant expertise. Also, it can be difficult to gather if the target is developed by several departments or by subcontractors. It can take time to identify who has the proper inputs or some industrial confidentiality problems may appear.

Two scales are proposed for this parameter:

Production of the input

- The inputs to be provided cannot be made available i.e. too complicated or too costly to produce, e.g. formal proof.
- The inputs are not naturally produced by the developer or have to be adapted to the evaluation needs (e.g. CC ST, specific evaluation rationales, etc.) and are thus only partially available before the evaluation. Important efforts need to be made to produce them.
- Majority of the required inputs are available and only partial adaptation or modification needs to be made to adapt them to the evaluation input requirements.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

- Production of the evaluation inputs is already fully integrated in the developer product life-cycle.

Gathering of the input

0- The inputs to be provided cannot be made available, e.g. due to confidentiality or strategic industrial issues.
1- The document to be provided are produced by actors requiring to face important administrative (e.g. several scattered development departments not used to exchange documents) or confidentiality issues (e.g. negotiated NDAs with tier ones and possibly their sub-contractors).
2- Scattered inputs needing time to be gathered.
3- Already easy to access required inputs.

2.1.3.8 Adaptation to ITS

The final parameter for comparison that we propose is the adaptation of an evaluation framework to ITS. Some evaluation frameworks simply don't work for ITS products. For instance, a framework such as FIPS 140-2 (NIST, 2002) which only evaluates cryptographic modules cannot evaluate the security of the OBU for example since most of the security functions provided by it are not cryptographic. Also, a generic framework like CC which may be used to evaluate an ITS product, is clearly not tailored for that; they have not been designed for the specific case of the ITS products or the automotive industry context (i.e. significant costs constraints, strict time-to-market constraints, large scale deployment, long life time products, etc.).

The scale we propose for that parameter is the following one:

Adaptation to ITS:

0. Cannot be used for ITS
1. Can be used but not adapted
2. Can be used and is adapted but can still can be further optimized for ITS
3. Fully optimized for ITS (no better solution)

2.2 Required investments

The main criticism of cyber-security evaluation schemes and more specifically for its main representative i.e. the CC, is their cost. We consider that the investment to be provided by the developer and/or the evaluation sponsor (when different) can be divided in four components. The main cost in the sense that it is the easiest one to identify is the price to be paid to the evaluator. But actually, other costs can be identified. The first one is the time, the time needed between the beginning of the evaluation and the end of it.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

When the developer has to wait to have its product evaluated in order to sell it or to be allowed to put it in an ITS vehicle, time becomes a highly critical parameter, even sometimes more important than money.

Also, evaluating a TOE may require other kind of costs and investments. In the state of the art the most demanding approach regarding the involved investments is the CC. For this specific approach we can note two evaluation activities that may require strong investments and that can be sometimes part of other frameworks. The first one which is quite specific to the CC and also integrated into SAF is the security of the developments. This implies that the developer ought to guarantee the physical and logical protection of the involved server(s) and terminal(s). If this is something already handled by the developer, then there are no specific investments to make, but regularly it takes significant time and money to secure the development environment. The most complicated and costly aspect is the physical protection, if not already present. Because it requires to invest on physical access control (e.g. cameras, locks, etc) which is costly and takes time to install. The software protection is also costly and takes time but it's usually easier to achieve than physical one.

Also, sometimes to justify some of the functional tests required by the CC but also from other approaches such as ISO 21434, full functional testing (and thus a specific testing platform) is required. For instance, this should be the case for many ITS components. Some ECUs need to be integrated into a test bench (that fully simulate the IVN) to function properly. And thus, to perform tests that cover all the interfaces and parameters, large and expensive tests bed need to be used. So, we point that cyber-security evaluation processes mays require different level of investments in equipment and infrastructure.

The second kind of investment that is found regularly in the CC (and in SAF) and other frameworks is the investment on specific expertise. For the CC, it is sometimes more efficient to hire someone with dedicated CC expertise. An expertise that will be used only for the evaluations but that becomes mandatory when certification is a real business concern. This expertise covers the knowledge and understanding of the global CC vocabulary, tracing and evidences requirements, etc. In case of higher-level evaluation, formal proofs can be required. This cannot be done by any developer and most of the time it requires specific dedicated profiles. This may imply to hire an external expert for it. So different kinds of specific expertise can be required by some evaluation frameworks and thus may imply various investments.

It is not trivial to evaluate the corresponding costs which may take different forms on a single scale; estimating accurately a monetary (or any other single cost value) lies outside our competences or the scope of this document. What we propose is a carefully-identified set of the involved costs which is already challenging. The final matrix result we propose might be hard to compare but finding a common scale to evaluate would be too restrictive.

So finally, we propose the following scales:

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

	Time Working days	Money Euros	Equipement Equirement requirement description	Expertise Expertise requirement description
Sponsor	Cost for the sponsor to gather input data Note: Most of the time the sponsor is the developer and then the cost			
Developers	Cost to produce (correct) assurance evaluation tasks inputs			
Evaluator	Cost to perform the evaluation tasks			

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

3 Existing evaluation approaches

In the deliverable 3.1 we have presented the state of the art of the IT security evaluation methodologies. Those methodologies are limited in number but include many references (i.e. instances of evaluation approaches) that can be classified under four main classes. We discuss them again briefly here. For each of them we will identify the *most used and recognized* propositions to be selected for the comparison with the SAF approach. The simple reason for that is the amount of feedback necessary to make our estimations. Even for the selected ones, the access to the necessary feedback to evaluate the different parameters and costs will be very limited and sometimes not sufficient.

3.1 Conformity checks

3.1.1 Methodology description

Conformity Checks (also called compliance assessment) is a form of evaluation that validates a product's or system's compliance to a specific reference. This approach needs to have a reference conformity list. This list has to be kept up to date and has to be relevant to the product type and its real needs in terms of functionality and security. There are two main limitations to the conformity check approach. First, the definition and maintenance of relevant conformity lists can be difficult or even infeasible in an industrial context (i.e. too many updates needed, no agreement on the conformity requirements, scope of conformance too restrictive, etc.). Also, anything not conformant to (a part of) the conformity list cannot be validated. On the other side, conformity checks provide usually the fastest and cheapest evaluation scheme compared to other methods, providing comparable levels of confidence. Also, the evaluation results are simple to understand and easily comparable since every test is known in advance and they are the same for every product evaluated.

The main certification (and thus evaluation) scheme that defines a normalized (completely defined in a formal manner) test suite suitable for Conformity Checks is the FIPS 140-2 standard (NIST, 2002). This certification process only concerns cryptographic products. The FIPS standards are public and developed by the United States federal government, aiming at ensuring some computer security and interoperability for the US governmental Information Systems.

Contrary to other frameworks, such as ITSEC, CC or the French CSPN (SOG-IS) (ANSSI) (ISO/IEC, 2009) (ISO/IEC, 2008) (ISO/IEC, 2008) FIPS evaluations do not need the specification of a security target. The list of functions and tests to be performed is directly defined by the FIPS 140-2 standard, which indirectly defines the security target together with the assurance component through the list of conformity checks.

In this approach, since the test requirements are defined in the standard, they age with it and the standard must be rewritten every time new security paradigms are required (i.e. new threats, new needs, etc.). For this reason, the FIPS 140-2 standard foresees to be reviewed every five years, whereas such a standard in the ITS world should be typically reviewed every 6 months considering the rapid evolution of the system. Also, even if cryptographic functions are quite well recognized and very limited in complexity and numbers, this is not the case when we consider the full implementation of an ITS architecture. Such architecture includes OSs, communication and security stacks, sensors,

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

applications and so on. Cryptographic functions are a very limited subset of those systems and scaling the methodology would be at least as expensive as developing the system themselves.

In many industrial sectors and when feasible, this scheme is the preferred one – see for example the Compliance Assessment process specified by the C-ITS Platform[1], the US Certification program for Connected Vehicles and the ETSI ITS validation platform for standardized protocols. But such an approach can only partially cover ITS security validation and so far, nothing close to a recognized and validated set of security requirements and their associated tests exists (despite the fact that this approach is regularly promoted).

3.1.2 Candidates for comparison

3.1.2.1 FIPS 140-2

A description of the FIPS has already been provided in the deliverable 3.1 section 2.2.4.

This US standard is the best example of a successful compliance assessment approach. Also, it only works for cryptographic modules. This certification however could be considered for some parts of the ITS-S. The best example being the C-ITS-S HSMs that should be used to provide the cryptographic functions required by the SAFERtec and more generally Day 1 ITS use cases (i.e. messages signature and encryption).

But if this standard helps to ensure that cryptographic functions work as defined, it does not provide strong evidences that the device resists to attacks, since no vulnerability tests are performed.

In what follows, we examine the FIPS 140-2 performance with respect to the earlier introduced evaluation parameters.

Evaluation tasks and assurance evaluation:

- FIPS consists of only one evaluation type, functional conformity. This is associated to functional tests.
- **Functional tests**

Quantifiable

- FIPS 140-2 evaluation result is a binary result of yes or no conformity is achieved for a specific functional scope (specific set of cryptographic functions). This can be associated to a nominal scale. Even if failed evaluation provides an order for product successfully certified and not certified once for the same scope, this ordering function is not applicable for two products certified for different sets of cryptographic functions.
- **1- Qualitative - nominal**

Reproducibility

[1] https://ec.europa.eu/transport/sites/transport/files/2017-09-c-its-platform-final-report.pdf

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

- The main advantage of the FIPS 140-2 approach is that the test suit is normalized and every evaluation runs the same tests. The results are thus fully reproducible.
- **5- fully reproducible**

Comparability

- The FIPS results are comparable on two levels. First level is the evaluation level, for the same scope two evaluations with different levels can be compared as one providing more assurance than the other (the one with the highest level). The second level of comparison is the set of cryptographic functions evaluated (different from each other or one being a superset of the other).
- **2- Fully comparable elements of proof for similar products (in terms of functionalities, e.g. VPNs, firewalls, etc.)**

Efforts needed to interpret evaluation results

- The results do not need any interpretation since for every test the formal expected result is provided. The results just need to be compared to the expected results.
- **3- Results need no interpretation**

Exhaustiveness

- The FIPS tests presents sets of cryptographic inputs (algorithms to be tested, keys, data input, etc.) and the expected results (encrypted data, signature, verification result, etc.). The test suites provided aims at testing at least once every parameter. However full exhaustiveness is not demonstrated nor achievable (the set of possible inputs is infinite).
- **1- Partial with no coverage evidences**

Level of recognition:

- FIPS is a US national organism, thus it is a national recognition even if it is known worldwide. Canada also officially recognizes certified products.
- **3- Officially recognized by several countries**

Level of maturity:

- Initial publication in 2001, thousands of certifications have been produced since then (https://csrc.nist.gov/projects/cryptographic-module-validation-program/validated-modules/search).
- **Scheme: 3- Used for decades (>10 years) on a large variety of products (>1000) by many evaluation labs (>30)**
- **Assurance activities: 3- Used for decades (>10 years) on a large variety of products (>1000) by many evaluation labs (>30)**

Assurance continuity:

- No specific approach known for re-evaluation. The certificate has no validity period since it always validates the conformity to the standard, no matter how the state-of-the-art changes.
- **Re-evaluation cost: 0- Evaluation costs reduction for new evaluation no matter the level of changes 0%**

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

- **Evaluation result expiration: 5- no limitation**

Evaluator expertise validation:

- All of the tests under FIPS 140-2 are handled by third-party laboratories that are accredited as Cryptographic Module Testing laboratories by the National Voluntary Laboratory Accreditation Program (NVLAP).
- **1- Evaluator expertise validated once by peers**

Independency of the actors:

- To our knowledge the financial dependency of the testing labs is not demonstrated.
- **1- Third party tests with no demonstration of independency of the evaluator regarding sponsor or certificatory**

Evaluation review:

- To our knowledge there is no external review of the laboratories' reports. Good practices almost certainly enforce internal review. Evaluation laboratories are notified and should probably have to demonstrate their quality processes, even if this is not known to us.
- **1. Internal review of the evaluation report**

Difficulty to gather expected element of proof:

- The only element of proof to provide is the TOE.
- **Production of the input: 3- Production of the evaluation inputs is already fully integrated in the developer product life-cycle.**
- **Gathering the input: 3- Already easy to access required inputs**.

Adaptation to ITS:

- FIPS 140-2 only addresses cryptographic modules. Thus, it only addresses a very limited part of ITS products or functions.
- **1- Can be used but not adapted**

Required investments:

Sponsor

The required investments for the sponsor are the cost of the evaluator and the time spent to hire the evaluator and manage the process. The SAFERtec consortium has some knowledge but not a deep insight of the exact prices and time associated to a FIPS certification, however we can provide a range we know is correct.

The certification time takes from 3 months up to one year for an average price of 40 to 80 thousand euros. The difference depends on the complexity of the product: difficulty to test its different interfaces and its number of interfaces; as well as the number of cryptographic functions to be evaluated.

Other than that, the only extra effort is to manage the certification request and the delivery of the TOE.

Developer:

All the costs are for the sponsor. No specific costs are identified for the developer. The only element of proof to provide is the TOE and its guidance.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

Evaluator:
On the evaluator side, no specific efforts are identified apart from the evaluation days which include: the installation of the TOE, the adaptation of the tests to the TOE interfaces (automated requests and response evaluation), report redaction.

FIPS	Time		Money Euros	Equipment requirement description	Expertise requirement description
	Working days	Elapsed period			
Sponsor	2	3 months to more than one year	40-80K€	TOE	NA
Developers	NA	NA	NA	NA	NA
Evaluator	20-80	NA	NA	NA	NA

3.2 Vulnerability tests

3.2.1 Methodology description

This approach simply defines an evaluation perimeter, not necessarily forming an actual and complete ST. Usually it only defines the product, the tests environment and associated limitations. Then an expert runs any tests of his/her choice during a predefined time on the defined scope. At the end, the result is the set of potential vulnerabilities identified by the tester. If no vulnerabilities are found, then the evaluation result states that the product resisted to an attacker during a number of days equals to the evaluation time.

Thus, this method allows validating the product's security level, providing low to medium assurance level. Also, on average the results are obtained faster than other methodologies; note that common tests take 20 to 30 days.

The problem with this methodology is that there is a great need of confidence in the tester competences. Also, results are not fully consistent or directly comparable since two testers are free to use completely different tests for the same product.

A formalized approach falling under this category is the French CSPN (ANSSI, 2014) where a detailed ST is required and the number of vulnerability test days is predefined i.e. 25 days for every product. This process is the only one that provides a certificate signed by the prime minister and recognized nationally.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

3.2.2 Candidates for comparison

3.2.2.1 CSPN

Evaluation tasks and assurance evaluation:

- The CSPN consists of only one evaluation activity: vulnerability analysis.
- **Vulnerability analysis**

Quantifiable

- The evaluation result is a technical report identifying if either there are or not vulnerabilities identified by the evaluator during the evaluation period for the security target scope (product configuration, operational environment assumptions, set of security functions to be evaluated). The result is mainly binary: yes or no. Even if in the case of failure, the reports describe the vulnerability found. Depending on the security target the same product can pass or fail the evaluation, if typically, vulnerability exists but is exploitable only under specific environment configurations. The results are nominal stating the conformity to one specific ST.
- **1- Qualitative - nominal**

Reproducibility

- The evaluation context is easily reproducible. For instance, the same evaluation can be run by two different evaluation laboratories that will follow the same process. However, even the evaluators are notified, their expertise may vary and the set of tests they run will depend on their own expertise. Without the evaluator, some tests if not fully detailed in the evaluation report will not be reproducible. Also, the tests produced are very dependent to the product API implementation (e.g. two different VPNs do not have the same interfaces). However, a minimum set of common tests is enforced by the ANSSI review when different evaluators tend to do too different tests suites. The tests are thus on many aspects reproducible to some extent even if always different. Even if parameter changes, more than half of the tests are done similarly over time and over the different evaluation labs. In the case where the evaluation report is available, as it shall describe every test and most of the test's steps and parameters, most of them can be replayed. We estimate that the overall level of reproducibility is slightly higher than 50%.
- **3- 60% reproducible**

Comparability

- Results are only comparable when the STs used for the evaluation are identical, or based on the same protection profile, meaning in both cases that the same set of security functions are evaluated under the same constraints. In that case what is comparable is the evaluation result i.e., if the evaluation succeeded or failed. Other than that, if STs are different, results cannot be compared, which is the case most of the time.
- **1- Provides elements of proof to be partially comparable for similar products (e.g. two firewalls, two OSs, two cryptographic modules)**

Efforts needed to interpret evaluation results

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

- The results need to be interpreted regarding the STs. When participating to these evaluations it is surprising to see how often a specific test result needs to be interpreted. For almost every evaluation there are one to several tests that need to be discussed by the evaluator and the certificatory (ANSSI) to validate if the result falls inside or outside of the scope of the ST. This is mainly due to the fact that STs are written in natural language that needs to be interpreted. One very typical example is the assumption made that the administrator is competent and trust worthy, and one test demonstrates that it is very easy to misconfigure the product and make it vulnerable, even for a trustworthy administrator. So, in such cases, it is hard to identify where to stop the competence requirements and where the product vulnerability due to poor implementation starts. The results provided are detailed enough to demonstrate a specific behavior and, in that sense, do not need interpretation. But determining if the observed behavior is or is not a vulnerability regarding the ST context requires experts' analysis and validation.
- **2- None subjective results which require security experts to be interpreted**

Exhaustiveness

- The CSPN evaluators have to demonstrate in their report that all interfaces and security functions identified in the ST have been tested.
- **2- Partial with demonstration of interface coverage**

Level of recognition:

- CSPN is an official French certification scheme defined by the ANSSI under the decree n° 2002-535. Even if other countries recognize the benefits of CSPN certified product, France is the only country where this certification is recognized and required.
- **2- Officially recognized by one country**

Level of maturity:

- The pilot evaluations started in 2012, since then hundreds of products have been certified and the evaluation process updated to better fit evaluation constraints (not all products can be evaluated with just one CSPN, re-evaluation accepted to be done in less than 25 days, etc.). The scheme uses the vulnerability analysis approach that is used also in CC or in more generic services provided for decades by cyber-security experts companies. Thus, the used assurance activity is fully mature.
- **Scheme: 2- Used for several years (>2) on a large set of products (>50) by several evaluation labs (>5)**
- **Assurance activities: 3- Used for decades (>10 years) on a large variety of products (>1000) by many evaluation labs (>30)**

Assurance continuity:

- No specific approach exists for re-evaluation. The validity of the certificate is not limited over time even if the ANSSI recommends using product certified within the last 5 years. However, a certificate maintenance process exists to produce a maintenance report based on an impact analysis produced by the developer and validated by a notified evaluator that changes in the

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

product do not have security impact and thus the product provides equivalent security to the certified version. But this is not a certificate.

- **Re-evaluation cost: 1- Evaluation costs reduction for new evaluation of less than 25% code changes > 25%**
- **Evaluation result expiration: 5- no limitation**

Evaluator expertise validation:

- Evaluators are notified by the ANSSI which evaluates regularly (every two years) the technical competences of the evaluators.
- **2- Evaluator expertise validated periodically (less than 2 years)**

Independency of the actors:

- In the notification process the ANSSI verifies the evaluation laboratory independence and impartiality in relation to developers for its evaluation activity.
- **2- Third party tests with demonstrated financial independency**

Evaluation review:

- All reports are reviewed by the ANSSI and discussed during a specific face to face meeting with the evaluator. Regularly extra tests are required by the ANSSI.
- **2- Third party review of the evaluation report**

Difficulty to gather expected element of proof:

- The only element of proof to provide is the TOE and its guidance documents.
- **Production of the input: 3- Production of the evaluation inputs is already fully integrated in the developer product life-cycle.**
- **Gathering the input: 3- Already easy to access required inputs**.

Adaptation to ITS:

- The CSPN certification scheme can be used for any IT products, thus including ITS products. The approach is one of the fastest to provide good level of assurance which is also something adapted to ITS domain.
- **2- Can be used and is adapted but can still be optimized for ITS**

Required investments:

Sponsor:
The required investments for the sponsor are the cost of the evaluator and the time spent to hire the evaluator and manage the process. The request for evaluation implies to fill in a short application form and contact an evaluation lab to define the evaluation details. Then the sponsor has to participate to a starting meeting with the evaluation lab and the ANSSI. The whole process takes slightly more than one day. The certification time takes between 3 to 5 months for an average price of 40 thousand euros.
Developer:
The mandatory elements of proof to provide are the TOE and its guidance. For products including cryptographic functions, which is the case of a large majority of them, the ANSSI requires a specific document that provides the cryptographic specifications and implementation details of the product. This document is usually about 10 to 30 pages long. We can estimate that it takes 2 to 3 days of work

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

for the developer to write it. Also, the developer might be involved in the TOE delivery and installation process, which is also about half to one day of work.
On average, we estimate that the evaluation process represents 3 days of work for the developer.
Evaluator:
On the evaluator side, no specific efforts are identified apart from the evaluation days which include: the installation of the TOE, the adaptation of the tests to the TOE interfaces (automated requests and response evaluation), cryptographic analysis, report redaction.

CSPN	Time		Money Euros	Equipment requirement description	Expertise requirement description
	Working days	Elapsed period			
Sponsor	1	3 to 5 months	40K€	TOE	NA
Developers	3	NA	NA	NA	NA
Evaluator	35	NA	NA	NA	NA

3.2.2.2 Generic third party's vulnerability tests

This evaluation approach is a generic service sold by IT security expert companies. It consists of unformalized vulnerability tests. Usually such service identifies a target such as an URL or a set/range of IP addresses. It validates with the sponsor the evaluation resources to be used and the tests limitation: evaluator profiles (expert, junior, etc.), number of working days for the service and technical means (tests from the internet, tests with the evaluator computer and tools connected locally, tests with the sponsor computers, etc.). From there, no further specifications are provided, only occasionally some documentation on the target is provided.

The evaluators then run all the tests they deem appropriate.

Evaluation tasks and assurance evaluation:

- As for the CSPN, it consists of only one evaluation activity: vulnerability analysis. But contrary to the CSPN the approach is not defined in a specific document and is only based on best practices. Nevertheless, the approach is de facto standardized by the state of the art.
- **Vulnerability analysis**

Quantifiable

- The quantifiable parameter is the same as the CSPN. The evaluation result is a technical report identifying if either there are or not vulnerabilities identified by the evaluator during the evaluation period for the security target scope (product configuration, operational environment assumptions, and set of security functions to be evaluated).

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

- **1- Qualitative - nominal**

Reproducibility

- The reproducibility is different from the one obtained by the CSPN approach, for two reasons. The first one is that there are no constraints on the evaluation reports content. For the CSPN the ANSSI defines and reviews mandatory content, including the description of all the tests steps, allowing to some extent the possibility for someone having access to the report to replay the tests. Here, there is no such guarantee. The second point is the harmonization of the test's activities implied by the evaluator's expertise validation and the report review. This standardization is limited de facto to the standardization of the approach which does not provide the same level of standardization. We estimate here that only 40% is reproducible thanks to de facto tests standards (e.g. finger printing, vulnerability scans, automated vulnerability exploitation with well-known dedicated tools such as nmap, nessus, Metasploit, burp, etc.).
- **2- 40% reproducible**

Comparability

- Compare results of vulnerability tests is very difficult. Even for similar products or systems, the degree of liberty left to the evaluator is too high to allow systematic comparisons. Also, the expertise of the evaluators can vary so much that it is really hard see impossible to know if equivalent tests run by different experts do provide the same results. Configuring properly testing tools, being able to develop correctly tests scripts or programs have a great impact on the observed results. So, comparing results of vulnerability in that context is at least hard, see impossible in many cases.
- **0- No comparison is possible between different evaluations**

Efforts needed to interpret evaluation results

- Here the challenge is that vulnerabilities are usually not defined for the TOE. In approaches where STs are used, vulnerabilities are defined by the ST requirements and context. In the nominal case of vulnerability tests, the owner of the tested TOE does not provide a risk analysis or the identification of the security objectives (based on identification of the assets to be protected) they want to reach. Thus, defining what vulnerability is for that specific TOE is left to the evaluator. So, the presented results are more subjective even if based on the state of the art and most of the time valuable empirical knowledge.
- **1- Subjective results that can be interpreted in different ways by different experts (with possible lack of consensus)**

Exhaustiveness

- The exhaustiveness depends on many factors and is almost never demonstrated. Here it depends on the evaluator's own evaluation processes, expertise and resources allocated to the evaluation. The resources limitation necessary implies coverage limitation. Also, the lack of process definition (unlike the CSPN) does not enforce any evaluation of test coverage.
- **1- Partial with no coverage evidences**

Level of recognition:

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

- The results provided by such services are not officially recognized or validated by countries. Even if the benefits of such activities are well recognized in a general manner, there are no official recognitions of the results provided but such commercial activities.
- **1- Existence of a community (public, academic or industrial) de facto adopting it by using it**

Level of maturity:

- Vulnerability tests are probably the oldest security validation approach. The maturity of this activity does not need to be demonstrated. Even if it's realization fully depends on the state-of-the-art evolution (which is the case for almost all evaluation approaches), the global process has been well experienced for decades on thousands of TOEs.
- **Scheme: 0 - Never used**
- **3- Used for decades (>10 years) on a large variety of products (>1000) by many evaluation labs (>30)**
- **Assurance activities: 0 - Never used**
- **3- Used for decades (>10 years) on a large variety of products (>1000) by many evaluation labs (>30)**

Assurance continuity:

- No assurance continuity is provided. A vulnerability test is only valid for the TOE state at the time of the tests. Even if the same evaluator can replay equivalent tests on the new TOE version, it is very limited. The variety of the targets and possible updates, as well as the lack of impact analysis process definition does not allow to say that systematic reuse of results and re-evaluation scope narrowing are possible. The time validity of an evaluation report is limitless. Only its results are limited by the elements aforementioned.
- **Re-evaluation cost:**
- **1- Evaluation costs reduction for new evaluation of less than 25% code changes > 25%**
- **Evaluation result expiration: 0 - less than 1 month**
- **5- No limitation**

Evaluator expertise validation:

- None.
- **0- No validation of the evaluator expertise by peers**

Independency of the actors:

- Even if most of cyber security experts are independent financially from their customers, this service does not require neither to be nor to demonstrate that they are.
- **1- Third party tests with no demonstration of independency of the evaluator regarding sponsor or certificatory**

Evaluation review:

- None is enforced, even if colleagues' reviews should be done as good practice.
- **1- Internal review of the evaluation report**

Difficulty to gather expected element of proof:

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

- Very few constraints exist. The only really required input is the TOE and should be easy to access.
- **Production of the input: 3- Production of the evaluation inputs is already fully integrated in the developer product life-cycle.**
- **Gathering the input: 3- Already easy to access required inputs**.

Adaptation to ITS:

- This approach is very flexible and cost effective. However, in SAFERtec we would argue that such an approach would not allow to enforce or guarantee security. In fact, there is no need or obligation to correct potential problems and no guarantee that the tests coverage is enough nor adapted. This evaluation process is too open to provide enough guarantees.
- **1- Can be used but not adapted**

Required investments:

Sponsor

The sponsor only has to hire and pay the evaluator. So, the main cost is the service, which consist on average of 10 to 20 days of work for prices going from 5K€ up to 20K€. The work can span over 1 month.

Developer:

The developer does not need to be implied directly in this evaluation process.

Evaluator:

Nothing besides the 10 to 20 working days.

Vulnerability Tests	Time		Money Euros	Equipment requirement description	Expertise requirement description
	Working days	Elapsed period			
Sponsor	0	1 month	5-20K€	TOE	NA
Developers	NA	NA	NA	NA	NA
Evaluator	10-20	NA	NA	NA	NA

3.3 Assurance framework

3.3.1 Methodology description

The Assurance framework approach is the most complete and exhaustive approach. It provides the highest assurance levels (i.e. level of confidence in the product security), but it is generally more expensive and time consuming. It also requires the involvement of rare and expensive accredited evaluators.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

The CC are inspired from two important assurance schemes appeared in United States and Europe: (defense, l985) and (SOG-IS).

The first version of the Common Criteria for Information Technology Security Evaluation, known as Common Criteria (CC) dates back to 1994 and the last version to be standardized (ISO/IEC, 2009) was released in 2009. Since then, regular revisions have been done but the global approach has not changed. The current version accessible on the common criteria portal (https://www.commoncriteriaportal.org/) and used for evaluations is the 3.1 Release 5.

It keeps the main concepts of ITSEC (SOG-IS): (i) the notion of the need of a proper ST target, (ii) the decomposition of the evaluation in generic evaluation tasks independent of any product or security requirements, (iii) the definition of several evaluation assurance levels, each providing a set of more stringent evaluation tasks and evidences requirements.

Eventually the CC provide a complete description and a reference set of security requirements to write formalized STs and the most extensive list of evaluation activities including any activities empirically recognized as having a potential impact on the final product security.

The CC global approach consists of the evaluation of every product life cycle elements that helps demonstrate that security requirements identified in the ST can be traced to the real product delivered to the end user. It proposes to evaluate the product life cycle management, the product architecture and full specification, the guides provided with the product to demonstrate that it can be easily used with the proper security configuration, the functional test run on the product and finally the vulnerability test to complete the whole assessment that the product fulfills the requirements stated in the ST and that those requirements cannot be bypassed. Vulnerability tests and conformity checks are included in the CC and are only subparts of a complete CC evaluation. No other methodologies cover so many aspects or are so well structured. That is why it is the best approach and accordingly the most expensive one. Also, it is the only one to benefit from an official international recognition agreement, officially signed by 31 countries (members, 2017).

3.3.2 Candidates for comparison

3.3.2.1 Common Criteria (CC)

The CC approach has been extensively described in the deliverable 3.1 section 2.2.3.

This approach is heavily criticized for its cost, duration and lack of flexibility (e.g. certification valid for only one version of the product, heavy administration latency and requirements etc.).

Evaluation tasks and assurance evaluation:

All evaluation tasks covered by the CC are presented in Table 2 and Table 3.

- **Security target evaluation**
- **Life-cycle**
- **Product specification and conception**

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

- **Functional tests**
- **Vulnerability analysis**
- **Guidance documents review**

Quantifiable

- The result of a CC evaluation is an assurance level together with a report stating if either the evaluation succeeded (the product is conformant to its security target) or failed and for which reasons. The CC evaluation results are thus quantifiable for the assurance aspect but not for the security aspect. As for CSPN and vulnerability tests the result is considered nominal.
- **1- Qualitative - nominal**

Reproducibility

- The CC try to normalize as many evaluation parameters as possible. Also, the CCRA and in Europe the SOG-IS try to harmonize as much as possible the evaluation activities in order to get the most uniform and reproducible evaluation scheme. Of course, the variety of products to evaluate and the level of complexity of the evaluation leave a large space of interpretation and liberty for the evaluators to do what they deem appropriate. This second part is the limitation to the reproducibility factor. Our estimation would be that 30 to 40 % of the evaluation tasks are sufficiently well defined to be reproducible and an extra 30 % are added by certification scheme harmonization activities.
- **3- 60% reproducible**

Comparability

- The normalized assurance level and ST structure including the security requirements help compare to some extent evaluations of similar products. The assurance level in the first place can be directly compared. Also, ST perimeters can be compared to some extent. If one ST contains strictly more SFRs then the evaluation scopes can be compared. One element defined by the CC that also helps to compare products is the definition of protection profiles. They help to identify products of the same functional type that provide equivalent security properties.
- **1- Provides elements of proof to be partially comparable for similar products (e.g. two firewalls, two OSs, two cryptographic modules)**

Efforts needed to interpret evaluation results

- As for the CSPN the evaluation results are quite precise and detailed in the evaluation reports, but some results still need to be interpreted. Even if STs are more structured, they still need to be interpreted for specific cases. It is the case mainly for the vulnerability analysis task but also for all the other evaluation tasks, even if it is less often.
- **2- None subjective results which require security experts to be interpreted**

Exhaustiveness

- One of the main efforts made by the CC is the tracing activities that forces the evaluator and the developer to evaluate as exhaustively as possible the TOE. Even starting from EAL one, all SFR and TSFI have to be verified. What varies is the depth of the verifications. For the highest

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

evaluation level (EAL 7) it even provides formal proofs of the security conformity of the product, but even if possible, it is rarely achieved.

- **2- Partial with demonstration of interface coverage**
- **Up to 3- Exhaustive - formal proof that all executions have been tested**

Level of recognition:

- The CC are the only framework officially recognized by several countries (members, 2017).
- **3- Officially recognized by several countries**

Level of maturity:

- The first standardization of the CC goes back to 1999 since then more than 3000 certificates have been produced (https://www.commoncriteriaportal.org/products/).
- **Scheme: 3- Used for decades (>10 years) on a large variety of products (>1000) by many evaluation labs (>30)**
- **Assurance activities: 3- Used for decades (>10 years) on a large variety of products (>1000) by many evaluation labs (>30)**

Assurance continuity:

- Assurance continuity is defined by the CC. A product update can be certified based only on an impact analysis study written by the developer and evaluated by an evaluation lab. In cases where the approach works i.e., the impact analysis is sufficient to demonstrate that updates have no security impact, the new certificate cost is only 10% of the original evaluation (usually it only involves the production of a document of 10 to 30 pages instead of several hundreds of pages of developers' inputs). But this approach works mainly for hardware products and only for limited updates' type. However, we can note that if a product is re-evaluated by the same laboratory, the average observed cost decrease (time and efforts to update the developer inputs and the evaluation reports) is about 50%. All obtained certificates assertions: products evaluated with no non-conformities observed by the evaluation laboratory at the evaluation date are limitless in their validity. However, the CCRA limits the recognition period to 5 years.
- **Re-evaluation cost**
- **2- Evaluation costs reduction for new evaluation of less than 25% code changes > 50%**

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

- **Up to 5- Evaluation costs reduction for new evaluation of less than 25% code changes > 90 %**
- **Evaluation result expiration: 0 - less than 1 month**
- **3- less than 5 years**

Evaluator expertise validation:

- Evaluation laboratories are audited and notified periodically for both procedure management conformant to the CC requirements and cyber-security expertise.
- **3- Quality and cyber-security expertise evaluated by peers (less than 2 years)**

Independency of the actors:

- The evaluation laboratories have to be accredited regarding the ISO 17 025. This requires the commitment and demonstration of independency of the labs.
- **2- Third party tests with demonstrated financial independency**

Evaluation review:

- National certification entities review all evaluation reports. The CCRA members harmonize the evaluation review and requirements.
- **3- Third party review of the evaluation report and harmonization of the reviews by several reviewing entities**

Difficulty to gather expected element of proof:

- The different evaluation inputs demand very important efforts and specific developments for the evaluation. All the documents have to trace elements of proof (indirectly) back to SFRs, which are specific to one evaluation. Some of those elements are confidential to the company which produces them and in the case of subcontractors' developments, they also have to provide those same elements. Thus, the process is really demanding and imposes the production of dedicated documents including very specific element of proof and possibly confidential data. In the specific context of ITS industry which is the one that we evaluate, we are typically in the case where data are produced by different independent companies, including sensitive and confidential industrial data.
- **Production of the input: 1. The inputs are not naturally produced by the developer or must be adapted to the evaluation needs (e.g. CC ST, specific evaluation rationales, etc.) and are thus only partially available before the evaluation. Important efforts need to be made to produce them.**
- **Gathering the input: 1- The document to be provided are produced by actors requiring to face important administrative (e.g. several scattered development departments not used to exchange documents) or confidentiality issues (e.g. negotiated NDAs with tier ones and possibly their sub-contractors).**

Adaptation to ITS:

- The CC provide very high level of assurance, which will be to our point of view required by future ITS developments (semi-automated and automated driving). The framework is also able to provide certification for any IT products, including ITS components. So, it can be used for ITS components. But it does not scale to large and complex systems such as the complete

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

car. It should only be used for its most sensitive components (e.g. OBU, ECUs). The main limitation to its adoption for ITS components is its cost, extensively discussed in section 4, not adapted to the automotive industrial constraints.

- **2- Can be used and is adapted but can still be optimized for ITS**

Required investments:
Discussed in section 4.

CC	Time		Money Euros	Equipment requirement description	Expertise requirement description
	Working days	Working days			
Sponsor	3	1y	87K	TOE	N/A
Developers	82	N/A	41K	Testing bench	CC input production
Evaluator	87	N/A	0	Testing bench	N/A

3.3.2.2 CARSEM

CARSEM has also been presented in deliverable 3.1 Section 3.

It is a first adaptation of the CC to the ITS domain. It provides enhancement by proposing to use CC evaluation methodology outside of the current certification scheme. This allows for three main enhancements:

- Use of several evaluating actors not all notified (cf. 2.1.3.4) which allows to parallelize evaluation tasks and lower third party's evaluator costs
- Not getting through a traditional certification scheme which makes the framework more flexible and induces less latency
- Use of continuous monitoring of development teams' activities which reduces the cost and time spent on ALC (Life-cycle support) evaluation tasks

Thus, since the evaluation requirements are the same, it is easy to confirm that the final assurance results are equivalent. And in the same time, the different enhancements can easily demonstrate a reduction of the evaluation duration and independent laboratory costs.

So, most of the assurance and evaluation scheme related parameters are identical to the CC. The only parameters that will change are the costs, evaluator expertise review, independency of the actors and the maturity of the framework.

<u>Evaluation tasks and assurance evaluation:</u>

- The main concept of CARSEM is to keep all the evaluation tasks and EALs defined by the CC. The evaluation activities are those of the CC.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

- **Security target evaluation**
- **Life-cycle**
- **Product specification and conception**
- **Functional tests**
- **Vulnerability analysis**
- **Guidance documents review**

Quantifiable

- The final result is the same report as for CCs.
- **1- Qualitative - nominal**

Reproducibility

- Also same as CC since evaluation task and product evaluation process are the same.
- **3- 60% reproducible**

Comparability

- Same as CC, since again the same evaluation tasks are used and even if all evaluators don't have their expertise reviewed by independent third parties, they will use well defined and validated quality processes to harmonize the results in the same way.
- **1- Provides elements of proof to be partially comparable for similar products (e.g. two firewalls, two OSs, two cryptographic modules)**

Efforts needed to interpret evaluation results

- Same as CC.
- **2- None subjective results which require security experts to be interpreted**

Exhaustiveness

- Same as CC.
- **2- Partial with demonstration of interface coverage**
- **Up to 3- Exhaustive - formal proof that all executions have been tested**

Level of recognition:

- The level of recognition is one of the major differences regarding assurance provided by CARSEM. The proposed framework is brand new and is used actually for the first time together with its enhancements in SAFERtec. This framework is currently promoted but not yet used by other actors.
- **0– No one recognizes the evaluation scheme besides the one who defined it**

Level of maturity:

- The proposed framework is brand new and is used actually for the first time together with its enhancement in SAFERtec. So, no feedback exists yet on the evaluation of products for CARSEM. But nevertheless, as evaluation tasks are taken from the CC, their level of maturity is identical to the one provided by the CC.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

- **Scheme: 0- Never used**
- **Assurance activities: 3- Used for decades (>10 years) on a large variety of products (>1000) by many evaluation labs (>30)**

Assurance continuity:

- CARSEM proposes the same mechanisms as the CC. However, the validity period of the evaluation results is not subject to the CCRA. There is no limitation to validity as long as no vulnerabilities are known for the product.
- **Re-evaluation cost:**
- **2- Evaluation costs reduction for new evaluation of less than 25% code changes > 50%** up to **5- Evaluation costs reduction for new evaluation of less than 25% code changes > 90 %**
- **Evaluation result expiration:**
- **5- no limitation**

Evaluator expertise validation:

- Here CARSEM proposes something different from the CC. In order to reduce the costs not all the tasks are done by CC notified evaluation lab. The less sensitive evaluation tasks (security target, specification, functional tests and guidance evaluations) are done by not notified but independent evaluators: the product integrators (the car manufacturer). These tasks already exist in the current industrial process (even if not following the CC requirements) and are indirectly validated again by the vulnerability analysis which is run by CC notified evaluation laboratories. In both cases the capabilities of the evaluators are either validated by third parties or by internal processes but they are validated.
- **3- Quality and cyber-security expertise evaluated by peers (less than 2 years)**

Independency of the actors:

- As for the previous factor, the less sensitive evaluation tasks (security target, specification, functional tests and guidance evaluations) are done by not notified but independent evaluators: the product integrators (the car manufacturer). These tasks exist in the current industrial process (even if not following the CC requirements) and are too important in terms of product quality to be overlooked or biased by only financial interests. Again, all those evaluations are indirectly validated by the vulnerability analysis which is run by CC notified evaluation laboratories.
- **2- Third party tests with demonstrated financial independency**

Evaluation review:

- There are no evaluation reviews proposed by CARSEM except from internal review required by any good quality process.
- **1- Internal review of the evaluation report**

Difficulty to gather expected element of proof:

- Same as CC.
- **Production of the input: 1- The inputs are not naturally produced by the developer or must be adapted to the evaluation needs (e.g. CC ST, specific evaluation rationales, etc.) and are**

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

thus only partially available before the evaluation. Important efforts need to be made to produce them.

- **Gathering the input: 1- The document to be provided are produced by actors requiring to face important administrative (e.g. several scattered development departments not used to exchange documents) or confidentiality issues (e.g. negotiated NDAs with tier ones and possibly their sub-contractors).**

Adaptation to ITS:

- CARSEM main objective is to adapt CC to the automotive industry. This adaptation manages to provide from our point of view equivalent assurance to CC evaluation but with important costs reduction (time, money).
- **2- Can be used and is adapted but can still be optimized for ITS**

Required investments:
Discussed and compared to CC and SAF in section 4

CARSEM	Time		Money Euros	Equipment requirement description	Expertise requirement description
	Working days	Working days			
Sponsor	3	5m	51K	TOE	N/A
Developers	40	N/A	20K	Testing bench	CC input production
Evaluator	36	N/A	0	Testing bench	N/A

3.3.2.3 SESIP

This recent approach was not identified in the deliverable 3.1 and thus not described there. This methodology has been developed and made public after the D3.1 work, in 2018 (TrustCB, 2018).

This approach is dedicated to IoT platforms. They define an IoT platform as the hardware/software providing an operating environment for an IoT Application. In their approach they propose to evaluate the different platform parts independently from each other. The composition i.e., the consistency of the different "local" security requirements when evaluating independently different platform parts is to be ensured by the verification of the different objectives on the environment.

Composition may occur between platform parts that are evaluated at different assurance levels. By default, the composed platform can claim at most the lowest assurance level of the platform parts it is composed of. The framework reuses the CC evaluation tasks but redefines new assurance levels named ITP1 to ITP5. Moreover, they redefine new ways to write and evaluate security targets (set of requirements to be evaluated). The main difference being that they provide a dedicated set of SFRs not written and evaluated the way that CC requires, but still evaluated.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

This framework is very recent and even if evaluations already took place, we are not aware of them. More generally, the only information and feedbacks available to us are the public ones. The justification and parameters evaluation will be for that reason summarised.

Evaluation tasks and assurance evaluation:

- To our knowledge same as CC even if some evaluation tasks are adapted (e.g. lightweight ST evaluation).
- **Security target evaluation**
- **Life-cycle**
- **Product specification and conception**
- **Functional tests**
- **Vulnerability analysis**
- **Guidance documents review**

Quantifiable

- Same as CC.
- **1- Qualitative - nominal**

Reproducibility

- To our knowledge this parameter is equivalent to CC for the same reasons. But the framework provides predefined elements for the IoT elements, so it could be slightly higher. We do not have enough elements to guarantee it.
- **3- 60% reproducible**

Comparability

- As for the previous parameter, to our knowledge this parameter is equivalent to CC for the same reasons but it could be slightly higher. We do not have enough elements to guarantee it.
- **1- Provides elements of proof to be partially comparable for similar products (e.g. two firewalls, two OSs, two cryptographic modules)**

Efforts needed to interpret evaluation results

- Same as CC.
- **2- None subjective results which require security experts to be interpreted**

Exhaustiveness

- Same as CC.
- **2- Partial with demonstration of interface coverage**
- **Up to 3- Exhaustive - formal proof that all executions have been tested**

Level of recognition:

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

- This framework is recognized by the TrustedCB foundation. This is an international consortium. However, the produced certificates do not seem to be recognized officially by any country.
- **1- Existence of a community (public, academic or industrial) de facto adopting it by using it**

Level of maturity:

- We are not aware of any public certification. But as for CARSEM the framework relies on CC evaluation tasks.
- **Scheme: 0- Never used**
- **Assurance activities:**
- **3- Used for decades (>10 years) on a large variety of products (>1000) by many evaluation labs (>30)**

Assurance continuity:

- Same as CC concerning re-evaluation. We are also not aware of any time limitation for the emitted certificate.
- **Re-evaluation cost:**
- **2- Evaluation costs reduction for new evaluation of less than 25% code changes > 50%**
- **Up to 5- Evaluation costs reduction for new evaluation of less than 25% code changes > 90 %**
- **Evaluation result expiration: 0 - less than 1 month**
- **5- no limitation**

Evaluator expertise validation:

- Only CC licensed labs are part of the scheme.
- **3- Quality and cyber-security expertise evaluated by peers (less than 2 years)**

Independency of the actors:

- Only CC licensed labs are part of the scheme.
- **2- Third party tests with demonstrated financial independency**

Evaluation review:

- To our knowledge there are no external reviews of the laboratories' reports, but we lack information on that matter.
- **1- Internal review of the evaluation report (?)**

Difficulty to gather expected element of proof:

- It requires the same inputs as CC evaluations since it runs the same evaluation tasks.
- **Production of the input: 1. The inputs are not naturally produced by the developer or must be adapted to the evaluation needs (e.g. CC ST, specific evaluation rationales, etc.) and are thus only partially available before the evaluation. Important efforts need to be made to produce them.**
- **Gathering the input: 1- The document to be provided are produced by actors requiring to face important administrative (e.g. several scattered development departments not used to**

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

exchange documents) or confidentiality issues (e.g. negotiated NDAs with tier ones and possibly their sub-contractors).

Adaptation to ITS:

- This framework is adapted to IoT. It is less generic than CC and is to our point of view less adapted than CC for ITS products evaluation. In fact, the framework proposed predefines architecture and security requirements for IoT products. They are not the same as the ones that we have identified specifically for ITS. If it may apply to some equipment's in the car, it does not apply to most or specific ITS elements. One of the main features of IoT being the requirement of internet connection whereas V2X communications pose different requirements. Actually, the work done to adapt the framework to IoT should be redone to change those adaptions to ITS requirements. What is interesting however in the study of that framework, is that it is an example of a CC adaptation done for one specific domain which seems to work.
- **0- Cannot be used for ITS**

Required investments:

We are not able to evaluate those parameters. Too many factors are unknown to us: price asked by the laboratories, type of tests (hardware tests?), ease of access and availability of the certification entities and evaluation laboratories, etc. We only expect them to be similar to CC evaluation, even if it is possibly slightly less.

SESIP	Time		Money Euros	Equipment requirement description	Expertise requirement description
	Working days	Working days			
Sponsor	Not known				
Developers					
Evaluator					

3.3.2.4 SAF

Here we present the different SAF's characteristics evaluation. Actually, regarding the extent to which SAF reuses CARSEM, a large number of SAF characteristics are identical to CARSEM. Differences appear in:

Evaluation tasks and assurance evaluation:

- SAF presents a new assurance family AOP, for operational vulnerability evaluation of systems including evaluated products for its most sensitive security functions. So SAF proposes a wider range of evaluation tasks.
- **Security target evaluation**
- **Life-cycle**
- **Product specification and conception**

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

- **Functional tests**
- **Vulnerability analysis**
- **Guidance documents review**
- **Operational evaluation.**

Quantifiable

- The final result is the same report as for CCs and CARSEM.
- **1- Qualitative - nominal**

Reproducibility

- Reproducibility is meant to be higher in SAF than CARSEM. All SAF enhancements, i.e. the tools proposed and the existence of the protection profiles aim to standardize as much as possible security functions, architectures and tests. So, SAF is meant to reach a higher level of reproducibility reaching (when attaining its full maturity) 80% in best cases.
- **3- 80% reproducible**

Comparability

- Also, comparability is meant to be higher thanks again to more standardized products and evaluation. But we don't think we can achieve full comparability. Product will still be different and some degree of liberty left to the evaluator, not allowing full comparability. This parameter should be equivalent to CARSEM.
- **1- Provides elements of proof to be partially comparable for similar products (e.g. two firewalls, two OSs, two cryptographic modules)**

Efforts needed to interpret evaluation results

- No enhancement proposed by SAF so same as CC and CARSEM.
- **2- None subjective results which require security experts to be interpreted**

Exhaustiveness

- No enhancement proposed by SAF so same as CC and CARSEM.
- **2- Partial with demonstration of interface coverage**
- **Up to 3- Exhaustive - formal proof that all executions have been tested**

Level of recognition:

- As for CARSEM, the proposed framework is brand new and is used actually for the first time together in SAFERtec. This framework is currently promoted but not yet used by other actors.
- **0– No one recognizes the evaluation scheme besides the one who defined it**

Level of maturity:

- Same as CARSEM.
- **Scheme: 0- Never used**
- **Assurance activities: 3- Used for decades (>10 years) on a large variety of products (>1000) by many evaluation labs (>30)**

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

Assurance continuity:

- No enhancement proposed by SAF so same as CC and CARSEM.
- **Re-evaluation cost:**
- **2- Evaluation costs reduction for new evaluation of less than 25% code changes > 50%**
- **Up to 5- Evaluation costs reduction for new evaluation of less than 25% code changes > 90 %**
- **Evaluation result expiration:**
- **5- no limitation**

Evaluator expertise validation:

- SAF re-uses CARSEM enhancements over CC and does not provide further ones for that characteristic.
- **3- Quality and cyber-security expertise evaluated by peers (less than 2 years)**

Independency of the actors:

- SAF re-uses CARSEM enhancements over CC and does not provide further ones for that characteristic.
- **2- Third party tests with demonstrated financial independency**

Evaluation review:

- SAF re-uses CARSEM characteristic.
- **1- Internal review of the evaluation report**

Difficulty to gather expected element of proof:

- Same as CC.
- **Production of the input: 1- The inputs are not naturally produced by the developer or must be adapted to the evaluation needs (e.g. CC ST, specific evaluation rationales, etc.) and are thus only partially available before the evaluation. Important efforts need to be made to produce them.**
- **Gathering the input: 1- The document to be provided are produced by actors requiring to face important administrative (e.g. several scattered development departments not used to exchange documents) or confidentiality issues (e.g. negotiated NDAs with tier ones and possibly their sub-contractors).**

Adaptation to ITS:

- CARSEM is a first adaptation to the automotive industry. SAF provides a further enhancement by providing dedicated tools and knowledge bases to fully adapt the framework to ITS products. The intent of SAF is to reach once fully mature the full adaptation and optimization to ITS.
- **2- Can be used and is adapted but can still be optimized for ITS**
- **Up to 3- Fully optimized for ITS (no better solution)**

Required investments:
Discussed and compared to CC and SAF in section 4

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

CC	Time		Money Euros	Equipment requirement description	Expertise requirement description
	Working days	Working days			
Sponsor	3	3	5m	39K	TOE
Developers	30	N/A	15K	Testing bench	CC input production
Evaluator	28	N/A	0	Testing bench	N/A

3.4 Security metrics and other security evaluation approaches

3.4.1 Methodology description

The aforementioned approaches in this section are the most commonly used ones. However, over the last three decades many researchers and practitioners have addressed the general problem of IT products validation, trying to introduce more specific and formalized approaches. So far, not fully satisfying (i.e. universal recognition with no cons) solution has been found (and it will probably never be).

To our knowledge the most comprehensive overview of the various efforts made on the evaluation and measurement of IT security domain was done 10 years ago by (Measuring Cyber Security and Information Assurance: a State-of-the-Art Report, 2009) and (Freiling, 2008). It covers software, standards ((ISO/IEC), (ISO/IEC)), taxonomies ((R. Vaughn, 2003), (Current trends and advances in information assurance metricsFredericton, 2004)), metric definitions ((Jaquith, 2007), (Freiling, 2008)), methodologies ((M. Howard, 2005), (Payne, 2001)), security databases ((Current trends and advances in information assurance metricsFredericton, 2004)), etc.

Their common goal is to demonstrate security properties by either modeling the target and justifying how the model guarantees the thwarting of modeled threats (formal proofs/software assurance tools (NIST, 2007), attack trees models (Clark, 2005) (Schneier, 1999)) . They provide means to compare and analyze security (e.g. attack surface measurements (Manadhata, 2006), security metrics (Willke, 2005)).

Since then no major paradigm shifting or revolutionary approaches have emerged, even if attempts are regularly made (Jianxin Li, 2012) (Samuel Paul Kaluvuri, 2013) (Ari Takanen, 2018) (ETSI, 2018).

They all face the criticism of security evaluation challenges ((Quality of protection: measuring the unmeasurable? , 2006), (On the brittleness of software and the infeasibility of security metrics, 2006)): relying on sole security expert's knowledge or being not adapted to rapidly evolving systems. And even

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

if works are still on-going and efforts are made to enhance evaluation methodologies, there are no newly proposed solutions and the same three main (aforementioned) approaches are used.

3.4.2 Candidates for comparison

One interesting approach regarding its level of recognition and the large industrial consortium participating to its development and promotion is the ETSI GS ISI 003. This work has been proposed by the R2GS club (https://www.gsdays.fr/Club-R2GS.html) which is composed of more than fifty large companies (banks, insurance, industries, cyber-security experts). This French club is associated to its equivalent clubs in Great-Britain, Germany, Italy and Luxembourg. They all together gather industrial knowledge on real attacks sharing their competences through common CERTS, SOCs and general threat intelligence. The approach they propose in the ETSI ISI 003 is based on real observation of current threats in the members systems. It proposes to address the event detection aspects of the information security processes in an organization, i.e. multi-site organisation generally conformant to best practices of IT systems deployments (LAN, DMZ, WAN behind firewalls, VPNs, anti-viruses, etc.). The maturity level assessed during event detection can be considered as a good approximation of the overall Cyber Defence and SIEM maturity level of an organization. For that they propose a set of Key Performance Security Indicators (KPSI) to be used for the evaluation of the performance of the overall security of the system. Those indicators shall provide assurance that the security configuration and counter-measures really counter the threats faced by the system. We provide in the following picture an example of the proposed KPSI. Other KPSI are proposed for: configuration monitoring, continuous software vulnerability assessment, user access and account monitoring, log collection analysis and archiving, etc.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

KPSIs	Levels (1 to 3)	Process/ People/ Tools	Applicable to the monitored perimeter of the whole organization	Applicable to SOC
1 Inventory of devices or software	1	Process	• Regulatory compliance • Perform regular scanning for unauthorized devices and software on the whole perimeter (against a continuously updated official list)	• Idem (applicable to the whole organization's perimeter asset inventory discovery)
		People	• Training on the importance to use only registered and managed devices and software (Cf. 70 % of all incidents due to not abiding by this basic rule), and to fight shadow-IT	• Idem (applicable to the SOC team)
		Tools	• Tools integrated to network and system management tools	• Idem (applicable to the whole organization's perimeter asset inventory discovery)
	2	Process	• Idem level 1	• Idem level 1
		People	• Idem level 1	• Idem level 1
		Tools	• Asset inventory discovery tools (active or passive - Cf. for example some IPS analyzing forbidden traffic) • Software inventory tools or file integrity checking tools to validate the list of authorized software (and version and patch level) has not been modified (applicable to each type of system, including servers, workstations, and laptops)	• Idem (applicable to the whole organization's perimeter asset inventory discovery)
	3	Process	• Idem level 1	• Idem level 1
		People	• Tie critical incidents and related vulnerabilities (likely if not managed) to their business impact (for example by building Information Protection Plans) to enhance users' concern and motivation	• Not applicable
		Tools	• Dynamic host configuration protocol (DHCP) server logging, and use of a system to improve the asset inventory and help detect unknown systems through this DHCP information	• Idem (applicable to the whole organization's perimeter asset inventory discovery)

Figure 1ETSI ISI 003 KPSI exemple.

Evaluation tasks and assurance evaluation:

- The tests proposed are only operational metrics.
- **Operational evaluation**

Quantifiable

- The goal of this approach is to provide quantitative metrics that represent real security or assurance scales. Even if the proposed KPSI are made to be on quantitative scales, we have already discussed (cf. section 2.1.2.1) that they are not real quantitative assurance metrics. They provide good quantitative assurance indicators, but not metrics. A greater value on the scale does not always imply a better assurance.
- **1- Qualitative - nominal**

Reproducibility

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

- The metrics are made to be reproducible. However, each system is different and the technical means to "reproduce" the KPSI evaluation are far from being easily reproducible. But it is at least the purpose of it, providing uniform and reproducible metrics. At a high-level definition, those metrics can be applied in any system. Only their instantiation is not so easy to reproduce.
- **3- 80% reproducible**

Comparability

- The metrics are made to be easily comparable with no interpretation. That's the main goal of it
- **2- Fully comparable elements of proof for similar products (in terms of functionalities, e.g. VPNs, firewalls, etc.)**

Efforts needed to interpret evaluation results

- The KPSI are indicators easy to understand. They shouldn't be misinterpreted and should not be considered as pure assurance metrics. But as indicators they are easy to understand. The consequences and actions to be taken to further evaluate the potential vulnerabilities or threat identified by those indicators are far more complicated to evaluate. Those require much higher level of expertise to interpret.
- **3- Results need no interpretation**

Exhaustiveness

- The exhaustiveness of the KPSI is not demonstrated. It really depends on their instantiation no matter the process followed.
- **1- Partial with no coverage evidences**

Level of recognition:

- This approach is followed by the RG2S members and their equivalent in other countries. There is no national recognition.
- **1- Existence of a community (public, academic or industrial) de facto adopting it by using it**

Level of maturity:

- This approach is used by tenth of companies worldwide. Even if the KPSIs are not evaluated by evaluation laboratories we consider the KPSI developers as such.
- **Scheme: 2- Used for several years (>2) on a large set of products (>50) by several evaluation labs (>5)**
- **Assurance activities: 2- Used for several years (>2) on a large set of products (>50) by several evaluation labs (>5)**

Assurance continuity:

- The concept is to provide continuous assurance by continuously evaluating and monitoring the KPSIs. Once implemented, they usually only need to be updated with the major system changes (updates of the firewall, anti-viruses, VPNs, OSs brands, etc.) most of the KPSIs

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

support minor system updates. The results provided by the metrics are usually valid for at most a few days or a couple of months, and then new measurements are made.

- **Re-evaluation cost: 5- Evaluation costs reduction for new evaluation of less than 25% code changes > 90 %**
- **Evaluation result expiration: 1- less than 6 months**

Evaluator expertise validation:

- The proposed approach is usually self-assessment based. There is no requirement on the validation of the expertise of those who run KPSIs in their system.
- **0- No validation of the evaluator expertise by peers**

Independency of the actors:

- Again, this approach is mainly self-assessment based.
- **0- Tests run by the developers themselves**

Evaluation review:

- To our knowledge there is no external review of KPSIs and their results.
- **0- No review of the evaluation report**

Difficulty to gather expected element of proof:

- The difficulty here is the implementation of the KPSIs. They are not always trivial to implement and sometimes for the most technical KPSIs, they are faced to the security policies and counter-measures in the system that limit the possibilities, e.g. for Cyber stress drills KPSI specific tools to stimulate attacks shall be run. This is not always compatible to security policies.
- **Production of the input:**
- **2- Majority of the required inputs are available and only partial adaptation or modification needs to be made to adapt them to the evaluation input requirements.**
- **Gathering the input:**
- **2- Scattered inputs needing time to be gathered.**

Adaptation to ITS:

- The purpose of the KPSIs is more IT system oriented than products, e.g., check periodically the level of security policy application regarding human practices (hygiene and compliant human behavior), training process to make employees aware of cyber risks, company security policy and main existing security measures (with focus on detection). So even if it could be used and provide interesting indicators it does not suit real ITS needs.
- **1- Can be used but not adapted**

Required investments:

It's really hard for us to estimate the average cost of such monitoring framework implementation. First of all, we do not have access to any concrete feedback from KPSIs developers. Second, the cost may greatly vary from users to users. The costs are proportional to the system complexity and to the resources the system owner is willing to invest in this approach, thus it can go from 10K€ and a few

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

weeks of implementation and deployment for small companies up to 100K€ and months. We will provide here a very rough estimation of what we would think it costs.

ETSI ISI 003	Time		Money Euros	Equipment requirement description	Expertise requirement description
	Working days	Elapsed period			
Sponsor	40	3 months	40K€	NA	NA
Developers	NA	NA	NA	Probes (IDS, IPS, scripts, etc.)	Assurance metrics
Evaluator	NA	NA	NA	NA	NA

3.5 General Best Practices developments

3.5.1 Methodology description

As mentioned in the previous sections, security assurance can be gain in many different ways. Some require the intervention of third parties, others only perform predefined sets of tests while others imply the use of operational tests. Many approaches try to define ways to demonstrate the security properties of products and do not scale to large systems. Most of them imply from the product integrator intervention of the developer, expert evaluators or other external third parties and only address products (component of limited size, well defined perimeters and supported by one main developer) and not systems (composition of products, e.g. the whole car, ITS central station).

Those approaches do not take into account global architecture designers and their industrial constraints: providing good assurance at system level in a cost effective and limited manner. This is exactly what the following approaches try to tackle. They do not define a product-centric approach, but a system approach and more specifically a car-level security validation. Those methodologies are specifically oriented toward car manufacturer needs; which is the case of the two following approaches we present here. They are both defining processes to fully integrate cyber-security assurance within product life-cycle management. They both define for each life cycle steps how to tightly manage and validate security requirements and implementation.

SAE J3061 aims at fulfilling the cybersecurity needs to be designed and built into cyber-physical systems throughout their development lifecycle to provide defence in depth. It covers not only design and development phases but also processes to monitor and respond to incidents in the field, and to address vulnerabilities in service and operation. It defines for each life-cycle phase (production, operation, service, and decommissioning) the required inputs, the processes to be used and the expected result. It provides information on some common existing tools and methods used when designing, verifying and validating cyber-physical vehicle systems as well as basic guiding principles on Cybersecurity for vehicle systems. The global approach is generic and very high level and its adaptation

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

to ITS can be further improved since it is not technology dedicated and is generic enough to be used almost straightforwardly for any IT complex system. But it is fully integrated in the automotive ecosystems and existing standards.

At the time of this deliverable writing the ISO/SAE 21434 is not finalized yet and is still under review process. It has not yet been adopted or fully used for commercial products. It follows the same principles and provides a similar solution as SAE J3061. However, each step inputs and outputs are from our point of view more detailed and more specific. Some (technical) examples are provided for each requirement and annexes propose even more lengthy and detailed examples. This helps the reader and future people in charge of the standard implementation to understand more precisely the requirements. Those requirements are decomposed as follow:

- Management of Cybersecurity (overall cybersecurity management, management during the concept phase and product development, during production, etc.)
- Risk assessment methods (asset identification, threat analysis, impact assessment, attack analysis, etc.)
- Concept Phase (cybersecurity relevance, initiation of product development, cybersecurity goals, etc.)
- Product development (system development phase, hardware & software development phase, **Verification and validation**, etc.)
- Production, operations and maintenance
- Supporting processes

For both approaches, we see that they define good overall management processes and best practices. But in both cases the drawbacks are that they are fully based on the car manufacturer expertise and risk analysis. No third-party review is mandatory, even if identified as a possibility by ISO 21434 and no expertise validation by peers are required. The whole process relies on the trust that the car manufacturer will have and will know what is required to secure systems and that their financial interests will not impact the process. Also implying that as car manufacturers they own the same cyber-security expertise as dedicated and internationally recognized experts. From our point of view, this is a risk that does not in the end help to obtain very high level of assurance.

3.5.2 Candidates for comparison

ISO/SAE 21434 is not yet published and has only been used in preliminary testing phase. Thus, very limited feedback exists and it is not accessible to us. The parameters evaluation that we present here are quite subjective and only based on the standard draft reading and our knowledge of equivalent framework or processes.

<u>Evaluation tasks and assurance evaluation:</u>

- It is hard to define the different evaluation tasks enforced by the standard, since most of them are optional and have to be defined by the car manufacturer. The main example being the penetration testing activity which is not mandatory but just proposed as a possibility among other testing activities in the annex E (Functional testing, Interface testing, Penetration testing, etc.). Again, the validation and assurance activities are fully left to the decision of the

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

car manufacturer regarding their risk analysis, e.g.: "for an item or component which is identified as cybersecurity-related a judgement, based on a rationale, shall be made to decide whether a cybersecurity assessment shall be performed. This rationale shall be documented and independently reviewed." So, assurance activities can (theoretically) go from nothing, up to CC like evaluation depending. Even if many requirements are made in terms of management (e.g.: A cybersecurity audit shall be performed to independently judge whether the organizational achieve the process related objectives of this document.) the detailed or expected content of the evaluation task is left to the car manufacturer decision.

- **Security target evaluation**
- **Life-cycle**
- **Product specification and conception**
- **Functional tests**
- **Vulnerability analysis**
- **Guidance documents review**
- **Operational evaluation**

Quantifiable

- There is not one final result. It is the composition of several processes output (risk analysis, specifications, tests, etc.). The final result is thus this collection of documents which represents a nominal result.
- **1- Qualitative - nominal**

Reproducibility

- The level of complexity of systems, the degree of freedom of interpretation of the required evaluation make the global approach hardly reproducible. Even if based on best practices the way to implement them are too numerous. Only very partial choices would be common from our point of view.
- **1- 20% reproducible**

Comparability

- Due to the great flexibility and level of freedom left to the evaluator, relevant evaluation tasks can take many forms. Also, the complexity of the system to be evaluated as well as their level of complexity handled so freely that do not help to compare the expected evaluation results.
- **0- No comparison is possible between different evaluations**

Efforts needed to interpret evaluation results

- There is a great effort required for the evaluator to determine on its own what to evaluate, how to evaluate and how to interpret the results. Even if, for all the evaluation tasks best practices shall be used. It is fully left to the evaluator's interpretation to decide what is a vulnerability and how to identify and test it.
- **1- Subjective results that can be interpreted in different ways by different experts (with possible lack of consensus)**

Exhaustiveness

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

- The exhaustiveness of the tests is not directly enforced e.g., the standard requires for functional tests the following: “Functional testing is applied to a component or system, possibly integrated in a test environment, to determine whether the functionality of the component or system meets the requirements. Exhaustiveness”. No specific methods or results are required to demonstrate the conformity to the product requirements. To our knowledge it is the same for all evaluation activities.
- **1- Partial with no coverage evidences**

Level of recognition:

- Yet the standard is not finalized, so its level of recognition does not go beyond the editorial working group. But current trends tend to show that work of the UNECE on international type approval regulation could use this standard as a reference.
- **1- Existence of a community (public, academic or industrial) de facto adopting it by using it**

Level of maturity:

- The standard is not finished. Schemes to apply it do not exist yet. But all required evaluation tasks use and recommend best practices approaches.
- **Scheme: 0 - Never used**
- **Assurance activities: 3- Used for decades (>10 years) on a large variety of products (>1000) by many evaluation labs (>30)**

Assurance continuity:

- Reused activities are mentioned in the standard. It consists of statements simply defining that in case of update the impact of the update and identification of “work product” that need to be updated consequently have to be assessed. It is really hard to evaluate the impact of such high-level recommendations without real feedback. There is no evaluation validity limit, however it is clearly required to regularly evaluate the impact of the state-of-the-art evolution on the system.
- **Re-evaluation cost: 1- Evaluation costs reduction for new evaluation of less than 25% code changes > 25%**
- **Evaluation result expiration: 5- no limitation**

Evaluator expertise validation:

- It does not go any further that the following requirement: “manage the competences and awareness needed to perform the cybersecurity activities;”. But no validations by peers are required.
- **0- No validation of the evaluator expertise by peers**

Independency of the actors:

- It consists mainly of self-assessment. If independent third parties can be implied, it is not made mandatory and pure self-assessment can be done.
- **0- Tests run by the developers themselves**

Evaluation review:

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement no 732319

- As stated for the previous parameter, no third-party intervention is required. It is required that cybersecurity rational shall be independently reviewed, but no third-party reviews are enforced.
- **1- Internal review of the evaluation report**

Difficulty to gather expected element of proof:

- Most of the required inputs are integrated in the existing product and systems' life-cycle of the different stakeholders involved. This is actually one of the objectives of those participating in the definition of this standard. So, if some new and specific elements of proof have to be developed, to our knowledge most of them already existing.
- **Production of the input: 2- Majority of the required inputs are available and only partial adaptation or modification needs to be made to adapt them to the evaluation input requirements.**
- **Gathering the input: 3- Already easy to access required inputs.**

Adaptation to ITS:

- This standard is designed to be specifically adapted and dedicated to automotive systems and thus ITS systems. However, the high-level requirements leave room to enhancement, since more detailed and technical descriptions of standard architectures and security requirement could be provided.
- **2- Can be used and is adapted but can still be optimized for ITS**

Required investments:
No feedback is yet available to us on the expected costs. We cannot evaluate those parameters.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

4 SAF and CC costs comparison

One important improvement aspect of the SAF comparison against other approaches is the cost gain compared to the normal CC evaluation on which it is based-on.

This section is dedicated to the cost analysis of SAF compared to the regular CC evaluation and to the CARSEM; the latter is the first enhancement of CC dedicated to ITS. We assess for each evaluation task the costs and time for regular CC, CARSEM and SAF corresponding to medium assurance level.

Common Criteria defines 7 assurance levels that are 7 sets of evaluation tasks. Since CARSEM and SAF propose to reuse the CC assurance tasks the same 7 assurance levels could be used. According to our recommendation we formalize 2 new sets of evaluation tasks (SAF1 and SAF2 as defined in the following table) that are different from the 7 initially proposed by the CC. Again, the reason being to tailor more efficient assurance activities (better ratio investments vs assurance, where CC main focus is assurance).

Assurance class	Assurance Family	Assurance Components by Evaluation Assurance Level								
		EAL1	EAL2	SAF1	EAL3	SAF2	EAL4	EAL5	EAL6	EAL7
DeVelopment ADV	ADV_ARC		1		1	1	1	1	1	1
	ADV_FSP	1	2	2	3	3	4	5	5	6
	ADV_IMP						1	1	2	2
	ADV_INT							2	3	3
	ADV_SPM								1	1
	ADV_TDS		1	1	2	2	3	4	5	6
Guidance Documents AGD	AGD_OPE	1	1	1	1	1	1	1	1	1
	AGD_PRE	1	1	1	1	1	1	1	1	1
Life-Cycle support ALC	ALC_CMC	1	2	-	3	3	4	4	5	5
	ALC_CMS	1	2	-	3	3	4	4	5	5
	ALC_DEL		1	1	1	1	1	1	1	1
	ALC_DVS				1	1	1	1	2	2
	ALC_FLR			1		3				
	ALC_LCD				1	1	1	1	1	2
	ALC_TAT						1	2	3	3
Security target Evaluation ASE	ASE_CCL	1	1	1	1	1	1	1	1	1
	ASE_ECD	1	1	1	1	1	1	1	1	1
	ASE_INT	1	1	1	1	1	1	1	1	1
	ASE_OBJ	1	2	2	2	2	2	2	2	2
	ASE_REQ	1	2	2	2	2	2	2	2	2
	ASE_SPD		1	1	1	1	1	1	1	1
	ASE_TSS	1	1	1	1	1	1	1	1	1
Tests ATE	ATE_COV		1	1	2	2	2	2	3	3
	ATE_DPT				1	1	1	3	3	4
	ATE_FUN		1	1	1	1	1	1	2	2
	ATE_IND	1	2	2	2	2	2	2	2	3
Vulnerability Assessment - AVA	AVA_VAN	1	2	2	2	3	3	4	5	5

Table 4 CC and SAF predefined assurance packages

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

For every evaluation activity, we will discuss the amount of effort in terms of working days for both the production of the input and the task evaluation needed for the SAF2 set of evaluation activities. Working days will be associated to specific expertise profiles for which we will provide estimated average person/day price. THOSE NUMBERS ARE EMPIRICAL KNOWLEDGE COMING FROM THE CONSORTIUM AND BASED ON HUNDREDS OF DISCUSSIONS AND SERVICES PROVIDED IN THE FIELD OF IT SECURITY EVALUATION. IT IS ALSO BASED ON OUR KNOWLEDGE OF WHAT ITS ACTORS ALREADY POSSESS AS EXPERTISE AND DOCUMENT PROCESS PRODUCTION. In fact, especially for ALC and ATE evaluation families, many standardized processes used by car manufacturers and their Tier-1 providers already require the development of such documents (e.g., quality process requirement of safety management enforced by standards such as ISO 26262 (ISO)). The concept of reusing existing processes and knowledge is fully part of the proposition and is thus estimated here.

WE CANNOT DETAIL HERE (FOR THE AVERAGE 30 TO 50 INPUTS DOCUMENTS AND THE ASSOCIATED THOUSAND PAGES THEY REPRESENT) ALL THE REASONS WHY FOR ALL TOE WE CAME UP WITH THESE EXACT FIGURES. WE PROVIDE FOR ALL TABLES HIGH LEVEL REASONS THAT ALLOWED US TO EVALUATE FOR AVERAGE PRODUCTS (SIZE, COMPLEXITY, ETC.) THE GAIN PROVIDED.

4.1 Security target evaluation (ASE)

SAF and CARSEM both recommend to only use standardized PP as it eases the writing but also the evaluation of the ST.

Since this evaluation does not require experts' competences, the car manufacturers using the product can directly validate the ST. In fact, this is beneficial as it won't require a high level of expertise and does not need to be done by accredited laboratories.

In what follows we present the corresponding efforts for this evaluation task. It is important to note that the reported values are an empirical estimation based on the consortium expertise, since no public studies exist on the matter.

We estimate a gain of 1 day over average PP instantiation since the PP is based on standardized elements and architecture, limiting the possible instantiations of SFRs, which is not the case for other PPs.

Assurance component name	Task Input	Regular CC efforts	CARSEM efforts	SAF efforts
ASE	Documents • Security Target	ST writing - Developer • 3 days to instantiate a PP • Extra efforts per evaluation task iteration: 0,5 days ST evaluation - ITSEF • 2 days • Extra efforts per evaluation task iteration: 0,5 days	ST writing - Developer • Idem CC ST evaluation – Car Manufacturer • Idem CC	ST writing - Developer • **2** days to instantiate the PP provided by SAFERtec • Extra efforts per evaluation task iteration: 0,5 days ST evaluation - ITSEF • **1** days

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

				• Extra efforts per evaluation task iteration: 0,5 days

Table 5 ASE costs

4.2 Life-cycle evaluation (ALC)

In the SAF framework as in the CARSEM approach, the **ALC class is not evaluated for one product but it is proposed to be evaluated every two years** by the involved development teams. Thus, the regular CC costs of ALC evaluation for one product here depends on the number of products evaluated over that period. We analyse this class with the underlying assumption of 5 products or versions (even if it may be much more). Here we choose to be conservative and consider less optimistic implementations of the framework. Thus, the presented **figures for one ALC evaluation** should be in the end **divided by 5.**

4.2.2.1 Life-cycle definition (ALC_LCD)

These inputs are generic for different products that could be developed by a team following the same identified life-cycle model. There are no restrictions on the possible procedures and tools to be used. Thus, there are also no restrictions identified for the document to describe them. They can be reused for other quality or organizational activities (e.g. safety management, security management, etc.).

Assurance component	Task Input	Regular CC efforts	CARSEM efforts	SAF efforts
ALC_LCD.1 Inputs production efforts	Documents: • Documents describing the different life-cycle stage management	Document production - Developer • Time needed to produce initial documentation: ○ 2 days • Extra efforts for the evaluation: ○ 0.5 days • Extra efforts for each evaluation task iteration: ○ 0,5 days	Document partially existing for the car manufacturer and tier-1 provider. Any extra effort corresponds to the evaluation.	Idem CARSEM
ALC_LCD.1 Evaluation efforts		Document evaluation - ITSEF • Iteration 1: 4 days • Iteration 2: 1 days • Higher iterations: 1 day Cost estimation: • Average Time: 6 days	Idem CC but only once every two years	Idem CC but only once every two years

Table 6 ALC_LCD costs

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

4.2.1 Development security (ALC_DVS)

The developer has to provide a detailed description for the following types of security measures:

- Physical protection of the development servers
- Procedural
  - granting and revocation of access rights to the development environment
  - roles and responsibilities in ensuring the continued application of security measures
  - admitting and escorting visitors to the development environment
- Logical protections on any development machines (e.g. servers, PC, laptops, etc.)

We have studied the already existing documents among SAFERtec partners for our estimation.

Assurance component	Task Input	Regular CC efforts	CARSEM efforts	SAF efforts
ALC_DVS.1 Inputs production efforts	Documents: • Documents describing the different security measures taken by the developer	Document production - Developer • Estimated time needed to produce initial security processes documentation: ○ 10 days • Extra efforts for the evaluation: ○ 2 days • Extra efforts for each evaluation task iteration: ○ 0,5 days	Document partially existing for the car manufacturer and tier-1 provider. The main efforts do not lie in producing the documents but achieving correct security. That is the case for most car manufacturers and their tier-1s. Only the extra efforts needed for the evaluation are required.	Idem CARSEM
ALC_DVS.1 Evaluation efforts		Document evaluation - ITSEF Evaluation days: • Iteration 1: 3 days • Iteration 2: 1 days • Higher iterations: 1 Cost estimation: • Average Time: 5 days	Idem CC but only once every two years	Idem CC but only once every two years

Table 7 ALC_DVS costs

4.2.2 Configuration Management capabilities (ALC_CMC and CMS)

Two different types of inputs are required. On one hand the developer has to provide the description of his configuration management processes and tools. And on the other hand, he shall provide the output configuration list produced by his configuration management for the full set of lists of elements he/she produced for the evaluation.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

SAFERtec partners have been interviewed to identify existence of equivalent processes and documents.

Assurance component	Task Input	Regular CC efforts	CARSEM efforts	SAF efforts
ALC_CMC.3 ALC_CMS.3 Inputs production efforts	Documents: • Documents describing the different version management processes and tools • Configuration list produced for the TOE and evaluation documents	Document production - Developer Process and documentation already existing. • Initial documentation: ○ 2 days • Extra efforts for the evaluation: ○ 1 day • Extra efforts for each evaluation task iteration: • 0,5 days	Document production - Developer Process and documentation already existing. Only extra efforts for the evaluation.	Idem CARSEM
ALC_CMC.3 ALC_CMS.3 Evaluation efforts		Document evaluation - ITSEF Evaluation days: • Iteration 1: 2 days • Iteration 2: 1 days • Higher iterations: 0,5 Cost estimation: • Average Time: 5 days	Idem CC but only once every two years	Idem CC but only once every two years

Table 8 ALC_CMC costs

4.2.3 Delivery (ALC_DEL)

For that evaluation task the developer shall provide documents describing:

- the delivery procedure
- how the receiver can verify the conformity and integrity of the delivered product (that it is indeed the one that has been certified)
- how the end user (here the car manufacturer on behalf of the future driver) who might not be the one the developer delivered his product to, can guarantee the integrity of the product.

SAFERtec partners have been interviewed to identify existence of equivalent processes and documents.

Assurance component	Task Input	Regular CC efforts	CARSEM efforts	SAF efforts
ALC_DEL.1 Inputs production efforts	Documents: • Documents describing the flaw reporting by the	Document production - Developer • Initial documentation: ○ 2 days	Document production - Developer Process and documentation already	Idem CARSEM

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

	user and the patch delivery	• Extra efforts for the evaluation: ○ 0,5 day • Extra efforts for each evaluation task iteration: ○ 0,5 days	existing. Only extra efforts for the evaluation.	
ALC_DEL.1 Evaluation efforts		Document evaluation - ITSEF Evaluation days: • Iteration 1: 1 days • Iteration 2: 0,5 days • Higher iterations: 0,5 Cost estimation: 1,5 working days for software and 2 for hardware	As final user evaluation included in the car manufacturer processes.	Idem CARSEM

Table 9 ALC_DEL costs

4.2.4 Flaw remediation (ALC_FLR)

The developer must provide references or evidences of an existing commercial commitment in handling flaw remediation.

They also have to describe flaw reporting, correction and patch distribution procedures.

SAFERtec partners have been interviewed to identify existence of equivalent processes and documents.

Assurance component	Task Input	Regular CC efforts	CARSEM efforts	SAF efforts
ALC_FLR.3 Inputs production efforts	Documents: • Documents describing the flaw reporting by the user and the patch delivery	Document production - Developer • Initial documentation: ○ 2 days • Extra efforts for the evaluation: ○ 0,5 day • Extra efforts for each evaluation task iteration: ○ 0,5 days	Document production - Developer Process and documentation already existing. Only extra efforts for the evaluation.	Idem CARSEM
ALC_FLR_.3 Evaluation efforts		Document evaluation - ITSEF Evaluation days: • Iteration 1: 2 days • Iteration 2: 1 days • Higher iterations: 0,5 Cost estimation: 3 working days	Idem CC but only once every two years	Idem CC but only once every two years

Table 10 ALC_FLR costs

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

4.3 Development (ADV)

4.3.1 Functional specification (ADV_FSP)

The functional descriptions do contain highly sensitive information. They mainly help to complete the full tracing proof to move from the SFRs (defined in the ST) to the product function and its interfaces implementing the SFR to be evaluated. The functional description here is at the interface level and must provide detailed descriptions of the interfaces including the protocols used and the interfaces' usage. For each interface, the security functions accessible through it shall be provided.

For each security function, the developer then describes:

- the purpose and the relevant SFR enforced (extract from the ST),
- interfaces and exchanged data,
- description of operations,
- logs and error messages,
- how to configure the function (parameters).

The description must also correspond to the description of the TOE actions described in the ST, with the corresponding subjects, objects and operations (ISO/IEC, 2009). The subjects and objects security attributes used for each operation must clearly appear in the functional architecture (data parameterizing the security function behaviour).

Thus, the required information is directly formatted for the evaluation. It is not a regular functional description of the product. Regarding the product and its use by the end user, some functional interfaces and functions might not be described since it's not mandatory (i.e. none SFR-related interfaces).

Assurance component	Task Input	Regular CC efforts	CARSEM efforts	SAF efforts
ADV_FSP.3 Inputs production efforts	Documents: • Documents describing the TOE interfaces and the tracing with the SFR (include TOE errors summary) • Errors summary for each TSFI invocations • Interaction with the non-security functions implementing part of the TOE	Document production - Developer. • Initial documentation: ○ 7 days • Extra efforts for the evaluation: ○ 2 days • Extra efforts for each evaluation task iteration: ○ 0,5 days	Developer Documentation usually partially existing. and has to be adapted to the evaluation. Only extra efforts for the evaluation.	Reduction of cost thanks to tools and methodology provided by WP2. Estimated time needed for **extra efforts** for the evaluation **reduced by 50% as SAFERtec provides templates.**
ADV_FSP.3 Evaluation efforts		Document evaluation - ITSEF Evaluation days: • Iteration 1: 2 days • Iteration 2: 1 days • Higher iterations: 0,5 Cost estimation:	Document evaluation – Car manufacturer Same efforts as CC	Better quality of inputs (more structured thanks to WP2 tools and methodology) should **reduce evaluation time by**

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

		3 working days		**30% thanks to the average increased quality and harmonized documents.**

Table 11 ADV_FSP costs

4.3.2 TOE design (ADV_TDS)

The developer must provide the decomposition of the TOE into smaller sub-systems and modules (i.e. the smallest functional entities in terms of design) in order to provide more details on how the TOE works and how security functions are implemented and decomposed in the TOE.

Thus, the documentation must describe the decomposition of the TOE and for each sub-system or module:

- Its purpose
- Its general behavior
- Its interfaces
  - Their specification (details of the operations executed)
  - The format of the input and output data
- Its interaction with other sub-systems/modules
- How it supports/implements TSFs

Assurance component	Task Input	Regular CC efforts	CARSEM efforts	SAF efforts
ADV_TDS.2 Inputs production efforts	Documents: • Documents describing the TOE sub-systems, their purpose, interfaces and the tracing with the SFR	Document production - Developer • Initial documentation: ○ 5 days • Extra efforts for the evaluation: ○ 0,5 days • Extra efforts for each evaluation task iteration: ○ 0,5 days	Developer Documentation usually partially existing. and has to be adapted to the evaluation. Only extra efforts for the evaluation.	Reduction of cost thanks to tools and methodology provided by WP2. Estimated time needed for **extra efforts** for the evaluation **reduced by 50% as SAFERtec provides templates.**
ADV_TDS.2 Evaluation efforts		Document evaluation - ITSEF • Iteration 1: 5 days • Higher iterations: 0,5 Cost estimation: • 6 working days	No enhancement.	Better quality of inputs (more structured thanks to WP2 tools and methodology) should **reduce evaluation time by 30% thanks to the average increased quality and harmonized documents.**

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

Table 12 ADV_TDS costs

4.3.3 Security Architecture (ADV_ARC)

The developer must provide documentation on the justification of the security architecture of its product and how it satisfies the SFR defined in the TOE. Thus, he must provide such information as:

- Security of the boot sequence
- Security domains (i.e. resources under the control of malicious entities) segregation
- Integrity of the TOE in operation (i.e. how measures assure the TOE integrity)
- How to by-pass security functions

The idea is for the developer to provide evidences that he understands the security design of its products and he knows how it counters the threat and possible attempt to bypass the security measures.

Assurance component	Task Input	Regular CC efforts	CARSEM efforts	SAF efforts
ADV_ARC.1 Inputs production efforts	Documents: • TOE security architecture justification	Document production - Developer • Initial documentation: ○ 5,5 days • Extra effort for the evaluation ○ 1 day • Extra efforts for each evaluation task iteration: ○ 0,5 days	Document production - Developer Documentation is usually not existing and has to be created for the evaluation. Idem CC.	Reduction of cost thanks to tools and methodology provided by WP2. Estimated time needed for **initial documentation** and **extra efforts** for the evaluation **reduced by 50% as SAFERtec provides architecture templates.**
ADV_ARC.1 Evaluation efforts		Document evaluation - ITSEF • Iteration 1: 3 days • Iteration 2: 2 days • Higher iterations: 0,5 days Cost estimation: 5 working days	Idem CC	Better quality of inputs (more structured thanks to WP2 tools and methodology) should **reduce evaluation time by 30% thanks to the average increased quality and harmonized documents.**

Table 13 ADV_ARC costs

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

4.4 Guidance documents (AGD)

4.4.1 Preparative procedures (AGD_PRE)

For these tasks, the inputs are both the TOE and the installation guidance. For cases where the installation is too complicated and the product is always integrated by the developer in the user operational environment then the regular installation procedure will be observed.

Assurance component	Task Input	Regular CC efforts	CARSEM efforts	SAF efforts
AGD_PRE.1 Inputs production efforts	Documents: • The implementation representation (code, models, etc.)	Document production - Developer Documentation already existing, has to be adapted to the evaluation. • Initial documentation: ○ 5 days • Extra efforts for the evaluation: ○ 0,5 day • Extra efforts for each evaluation task iteration: ○ 0,5 days	Document production - Developer Documentation already existing, has to be adapted to the evaluation. Only extra efforts for the evaluation.	Idem CARSEM
AGD_PRE.1 Evaluation efforts		Document evaluation – ITSEF • Iteration 1: 2 days • Iteration 2: 0,5 days • Higher iterations: 0,5 days Cost estimation: 2,5 working days	Document evaluation – Car manufacturer Same efforts as CC	Idem CARSEM

Table 14 AGD_PRE costs

4.4.2 Operational user guidance (AGD_OPE)

Operational user guidance should describe the security functionality provided by the TSF, provide instructions and guidelines (including warnings), help understand the TSF and include the security-critical information, and the security-critical actions required, for its secure use. Misleading and unreasonable guidance should be absent from the guidance documentation, and secure procedures for all modes of operation should be addressed. Insecure states should be easy to detect.

The CC make requirements on the content of the guidance that are mandatory to fulfil either directly (dedicated CC justification sections) or not (information present in the documents but not identified as CC requirement fulfilment):

- List of available functionalities with “warnings” in case of security functions

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement no 732319

- How to use the interfaces (e.g. commands or GUI)
- List of parameters that the (user) role can modify and "secure values" (since not all parameters values fall under the CC certification scope)
- Identification of all modes (incl. after failure) and associated procedures
- All required procedures (i.e. objectives for the environment in the security target) must be present in the guidance

Thus, the expected guidance provided for the evaluation is clearly evaluation-oriented and must contain a lot of elements that do not appear in regular guidance. Even if the TSF shall not be explicitly declared as such in the document they still have to be clearly detailed. The dependency of the evaluation family to ADV_FSP provides the guarantee that the evaluator can identify the TSF behind each interface. Good and exhaustive guidance should be enough even if not written for the evaluation.

Assurance component	Task Input	Regular CC efforts	CARSEM efforts	SAF efforts
AGD_OPE.1 AGD_OPE.1 Evaluation efforts	Documents: • Operational guidance (user, admin, config, etc.)	Document production - Developer • Initial documentation: ○ 5 days • Extra efforts for the evaluation: ○ 0,5 day • Extra efforts for each evaluation task iteration: ○ 0,5 days Document evaluation – ITSEF • Iteration 1: 2 days • Iteration 2: 0,5 days • Higher iterations: 0,5 days Cost estimation: 2,5 working days	Document production - Developer Documentation already existing, has to be adapted to the evaluation. Only extra efforts for the evaluation needed. Document evaluation – Car manufacturer Same efforts as CC	Idem CARSEM Idem CARSEM

Table 15 AGD_OPE costs

4.5 Tests (ATE)

4.5.1 Functional tests (ATE_FUN)

The developer must provide its test plan including tests scenarios or test scripts. For each scenario, the developer must describe the prerequisite, operations and expected results.

The "real" tests result for the TOE must also be provided. The test documentation and results shall justify the coverage of the TSFI identified in ADV_FSP. Thus, the TSFI in ADV_FSP must appear directly or indirectly, but if indirectly then the results rely on the fact that the evaluator has access to the ADV_FSP documents and evaluation.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

Assurance component	Task Input	Regular CC efforts	CARSEM efforts	SAF efforts
ATE_FUN.1 Inputs production efforts	Documents: • Test plan • Test results	Document production - Developer • Initial documentation: ○ 4 days • Extra efforts for the evaluation: ○ 2 days • Extra efforts for each evaluation task iteration: ○ 0,5 days	Document production - Developer Documentation already existing, has to be adapted to the evaluation. Only extra efforts for the evaluation are needed.	The D3.2 presents assurance metrics to quantify the trustworthiness attributes of the Connected Vehicle System. Those can be used as a basis for the test plan definition. The AF Toolkit provides the functionality (see D6.1, D6.2) to associate the main ITS interfaces with a set of proposed tests to serve the purposes of the ATE class. Estimated time needed for **initial documentation** and **extra efforts** for the evaluation **reduced by 50% thanks to templates provided by SAFERtec.**
ATE_FUN.1 Evaluation efforts		Document evaluation • Iteration 1: 3 days • Iteration 2: 2 days • Higher iterations: 0,5 days Cost estimation: 5 working days	Document evaluation – Car manufacturer Same efforts as CC	The evaluators will benefit from the same tools and the implied de facto standardization of the evaluation task. Estimated **evaluation time reduced by 30% thanks to the average increased quality and harmonized documents.**

Table 16 ATE_FUN costs.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

4.5.2 Coverage (ATE_COV and ATE_DPT)

For these evaluation tasks, there is no expectation for an exhaustive test coverage of every possible behaviour. What must be provided are evidences that all TSFI have been tested and all subsystems too.

The developer thus provides the tracing of his tests to the TSFI and the sub-systems of the TOE (described in ADV_FSP) and thus demonstrates that they are all covered by at least one test.

Assurance component	Task Input	Regular CC efforts	CARSEM efforts	SAF efforts
ATE_COV.2 ATE_DPT.1 Inputs production efforts	Documents: • Presenting the tracing of tests to TSFI and subsystems	Document production - Developer • Initial documentation: ○ 5,5 days • Extra efforts for the evaluation: ○ 0,5 day • Extra efforts for each evaluation task iteration: ○ 0,5 days	Document production - Developer Documentation not existing, has to be created for the evaluation. Idem CC	The D3.2 presents assurance metrics to quantify the trustworthiness attributes of the Connected Vehicle System. Those can be used as a basis for the test plan definition. The AF Toolkit provides the functionality (see D6.1, D6.2) to associate the main ITS interfaces with a set of proposed tests to serve the purposes of the ATE class. Estimated time needed for **initial documentation** and **extra efforts** for the evaluation **reduced by 50% thanks to templates provided by SAFERtec.**
ATE_COV.2 ATE_DPT.1 Evaluation efforts		Document evaluation - ITSEF • Iteration 1: 3 days • Iteration 2: 2 days • Higher iterations: 0,5 days Cost estimation: 5 working days	Document evaluation – Car manufacturer Same efforts as CC	The evaluators will benefit from the same tools and the implied de facto standardization of the evaluation task.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

				Estimated **evaluation time reduced by 30% thanks to the average increased quality and harmonized documents.**

Table 17 ATE_COV and DPT costs

4.5.3 Independent testing (ATE_IND)

The required input for this task is just the TOE. The dependencies of this task imply that the evaluator has also the TOE correctly installed (AGD_PRE).

The independent tests are done on the basis of the information provided in ATE_FUN and the analysis of the coverage which implies an indirect dependency to ADV class.

Assurance component	Task Input	Regular CC efforts	CARSEM efforts	SAF efforts
ATE_IND.2 Inputs production efforts	Documents: • TOE	Document production - Developer Average Time: 1 days (TOE delivery).	Idem CC	Idem CC
ATE_IND.2 Evaluation efforts		Document evaluation - ITSEF • Iteration 1: 8 days • Iteration 2: 2 days • Higher iterations: 1 days Cost estimation: • 10 working days	Document evaluation – Car manufacturer Same efforts as CC	The evaluators will benefit from the same tools and the implied de facto standardization of the evaluation task. Estimated **evaluation time reduced by 30% thanks to better test reuse provided by SAFERtec harmonization.**

Table 18 ATE_IND costs

4.6 Vulnerability assessment (AVA)

Here no specific evidences are required for the developer to provide besides the TOE, so we do not identify any impact on the development process. However, the same remark can be done as for low

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

level evaluation and development cost can be greatly increased even if we don't include them in the evaluation inputs development impact calculation.

Assurance component	Task Input	Regular CC efforts	CARSEM efforts	SAF efforts
AVA_VAN.3 Inputs production efforts	Documents: • TOE	Document production - Developer • Up to 100% costs increase compared to implementation without security functions and 50% on average (based on empirical observations)	Document production - Developer Idem CC	Same enhancement as for ATE. Tools and methodologies developed in WP3 and 6 will provide reference standards and architecture, clearly identified security requirements that will minimize the developer studies to implement a secure product. Estimated **development time for security functions reduced by up to 30%.**
AVA_VAN.3 Evaluation efforts		Document evaluation - ITSEF • Iteration 1: 20 days • Iteration 2: 3 days • Higher iterations: 1 days Cost estimation: VAN, 25 working days	Idem CC	More standardized product and security functions as enforced by WP3 requirements and WP6 tools and data bases will help to provide predefined tests suites and ease security tests. Estimated **vulnerability test time reduced by up to 20%.**

Table 19 AVA_VAN costs

4.7 Total efforts and costs

Summing up all the above efforts we obtain the following totals (Table 22 and Table 23).

Let's note that for all CARSEM evaluation iterations, the involved costs (and the extent to which they are similar) have been presented in (CARSEM: A Cooperative Autonomous Road-vehicles Security Evaluation, 17-21 September 2018); the parallelization of tasks could imply a higher number of iterations for all evaluation tasks except ASE (which do not depend of other evaluation tasks). This is also valid for SAF since the same principles apply.

So, for all CC cost, we add an extra iteration in the following totals.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

Cost Inputs		ALC_LCD	ALC_DVS	ALC_CMC ALC_CMS	ALC_DEL	ALC_FLR	ASE	ADV_FSP	ADV_TDS	ADV_ARC	AGD_PRE	AGD_OPE	ATE_FUN	ATE_COV ATE_DPT	ATE_IND	AVA_VAN	Total efforts
Developer	CC	2	13	4	3,5	3,5	4	10	6,5	7,5	6,5	6,5	7	7	1		82
	CARSEM	1	6,5	1,5	2	2	4,5	3,5	2	8	2	2	3,5	7,5	2		48
	SAF	1	6,5	1,5	2	2	3,5	3,5	2	3,75	2	2	2	4	2		37,8

Table 20 Total efforts for input production: CC, CARSEM, SAF

Evaluation costs		ALC_LCD	ALC_DVS	ALC_CMC ALC_CMS	ALC_DEL	ALC_FLR	ASE	ADV_FSP	ADV_TDS	ADV_ARC	AGD_PRE	AGD_OPE	ATE_FUN	ATE_COV ATE_DPT	ATE_IND	AVA_VAN	Total efforts
Sponsor (car manufacturer)	CC																0
	CARSEM						3,5	3,5			3	3	4	5,5	11		33,5
	SAF						1,5	2,5			3	3	4	4	8		26
CC or SOG-IS approved ISO 17025 ITSEF	CC	6	5	5	1,5	3	3	3	6	5	2,5	2,5	5	5	10	25	87,5
	CARSEM	1,4	1,2	1,1	0,4	0,7			6,5	5,5						25	38,0
	SAF	1,4	1,2	1,1	0,4	0,7			4,5	4						20	29,5

Table 21 Total efforts for evaluation tasks: CC, CARSEM, SAF

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

From internal consortium knowledge:

- the average price per day for an ITSEF evaluator is about 1000 €
- the average price per day for an internal car manufacturer engineer is about 750 €

The second price should actually be overestimated (on purpose). Actually, from what we know it should be closer to 500€ for average engineer working day. But we expect more senior profile including some expertise on CC formation. Because even if there is no extra technical expertise need for identified CC evaluation tasks, there is a need of specific training to understand CC formalism and reports productions. It's probably over estimated but we do not want to underestimate it and provide false sense of overestimation of our proposition.

The final monetary cost is the following.

Cost Inputs		Total costs
Developer	CC	61500
	CARSEM	30375
	SAF	22687,5

Table 22 Total costs in euros of input production

	Evaluation costs		Total
	Sponsor (car manufacturer)	CC or SOG-IS approved ISO 17025 ITSEF	
CC	0	87500	87500
CARSEM	22125	36820	58945
SAF	16500	28320	44820

Table 23 Total evaluation activities cost in euros

5 Conclusions

Comparing cyber-security evaluation methods is difficult. Both the lack of publicly available data on evaluation results and formal parameters to compare the different approaches make such studies challenging. In that context, the demonstration of SAF matching the ITS assurance needs is difficult. However, in this document we have managed to propose a large list of formalized parameters (having an impact on the confidence of cyber-security evaluation results) allowing comparisons. From our point of view, even if the proposed scales to evaluate those parameters can be subject to criticism or further refinement, those greatly help us provide common grounds for discussion. It helps to better formalize any discussion or arguments on how to define and justify the suitability of one specific approach.

When trying to identify an appropriate or the most efficient cyber-security evaluation scheme (either in a specific context or not), we are clearly in a case of trade-off and not in a case of one solution being

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

known as better than all others. This is clearly reflected in both Table 24 Final comparison table: assurance characteristics and Table 25 Final comparison table: cost. Those tables help to identify that all approaches have pros and cons. None of them is better than the other on all parameters, even not SAF.

One very important point for us in this study is to stress out that the state of the art demonstrates that high levels of confidence cannot be achieved without large amounts of efforts. Security problems lie in the very details of each specific implementation. So, to obtain assurance and being sure that all those details have been evaluated, necessarily large amount of work has to be done. Since no approach mastered to demonstrate full exhaustivity or never demonstrated to be widely applicable (formal proofs are still so far too complicated to be widely applied), we have to choose among the non-exhaustive approaches the most efficient and adaptive one.

In that context, in this document, we have demonstrated that the assurance framework that we propose is neither the best nor the worst (all green or red parameters, cf Table 24 and Table 25), but an approach that we think has been adapted to the ITS domain needs. In fact, we have identified that ITS systems' needs will require a flexible framework providing possibly high assurance level (several possible levels of assurance). ITS security challenge will go from ensuring drivers data privacy in open informative systems up to securing data of safety-critical autonomous driving application controlling the vehicle physics (speed, heading, breaks, etc.) implying possible risks on life and physical damages. However, the framework to be used has to be economically efficient and viable and should not stop the ITS deployment by requesting too high delays and monetary investments for the developers.

From our study we finally identify that SAF:

- has amongst the highest level of assurance thanks to the reuse of CC evaluation tasks
  - Largest set of evaluation task (larger than CC)
  - High level of maturity of the proposed evaluation tasks
  - Ensures high level of assurance and exhaustiveness of the evaluation
  - Independency and validation of the evaluator expertise
  - Higher than FIPS, CSPN, Vulnerability analysis, ETSI GS 003, ISO SAE 21 434
- provides good assurance continuity and real adaptation to ITS
  - Standardization and tools proposed for the developer inputs production
  - Better adaptation than the regular CC evaluations implying also faster re-evaluations
- is an affordable framework
  - The less expensive framework amongst the highest assurance framework (CC and CARSEM)
  - Only 10 to 20 % more expensive than FIPS, CSPN, Vulnerability analysis
  - Expected to be half the cost of regular CC evaluations (in terms of time, money, required expertise, equipment, etc.)
- still has the following drawbacks
  - Lack of international recognition (that CC provides)
  - Difficulty to gather inputs (similarly to the CC evaluation)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

So, for all these reasons we think that a framework providing dedicated tools and knowledge bases to standardized security counter-measures developments and assurance proofs is the best approach, since it manages to provide high assurance for lower costs. It is exactly why and how we designed SAF.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

	Evaluation tasks and assurance evaluation	Quantifiable	Reproducibility	Comparability	Efforts needed to interpret evaluation	Exhaustiv.	Level of recognition	Level of maturity		Assurance continuity		Evaluator expertise validation	Independency of the	Evaluation review	Difficulty to gather		Adaptation to ITS
								Scheme	Assurance	Re-evaluation	Evaluation result				**Production of the input**	**Gathering of the input**	
FIPS 140-2	Functional tests	1	5	2	3	1	3	3	3	0	5	1	1	1	3	3	1
CSPN	Vulnerability analysis	1	3	1	2	2	2	2	3	1	5	2	2	2	3	3	1
Vuln. A	Vulnerability analysis	1	2	1	1	1	1	3	3	1	5	0	1	1	3	3	1
SESIP	Security target evaluation Life-cycle Product specification and conception Functional tests Vulnerability analysis Guidance documents review	1	3 (4?)	1	2	2 and up to 3	1	0	3	3 up to 5	5	3	2	1	1	1	0
ETSI GS ISI 003	Operational evaluation	1	3	2	3	1	1	2	2	5	1	0	0	0	2	2	1
ISO SAE 21434	Security target evaluation Life-cycle Product specification and conception Functional tests Vulnerability analysis Guidance documents review Operational evaluation	1	1	0	1	1	1 possibly 3 in the future	0	3	1	5	0	0	1	2	3	2
CC	Security target evaluation Life-cycle Product specification and conception Functional tests Vulnerability analysis Guidance documents review	1	3	1	2	2 and up to 3	3	3	3	2 up to 5	3	3	2	3	1	1	2
CARSEM	Security target evaluation Life-cycle Product specification and conception Functional tests Vulnerability analysis Guidance documents review	1	3	1	2	2 and up to 3	0	0	3	2 up to 5	5	3	2	1	1	1	2
SAF	Security target evaluation Life-cycle Product specification and conception Functional tests Vulnerability analysis Guidance documents review Operational evaluation	1	4	1	2	2 and up to 3	0	0	3	2 up to 5	5	3	2	1	1	1	3

Best value obtained by a framework
Worst value obtained by a framework
Not comparable values

Table 24 Final comparison table: assurance characteristics

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

	Sponsor					Developers					Evaluator				
	Time		Money	Equipement	Expertise requirement	Time		Money	Equipemen	Expertise requiremen	Time		Money	quipemen	Expertise requirement
	W.D.	Elapsed				W.D.	Elapsed				W.D.	Elapsed			
FIPS 140-2	2	3m -> 1y	40-80K€	TOE	N/A	N/A	N/A	N/A	N/A	N/A	20-> 80	N/A	N/A	N/A	N/A
CSPN	1	3m -> 5m	40K€	TOE	N/A	3	N/A	N/A	N/A	N/A	35	N/A	N/A	N/A	N/A
Vuln. A	0	1m	5-20K€	TOE	N/A	NA	NA	NA	NA	NA	10->20	NA	NA	NA	NA
SESIP	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ETSI GS ISI 003	40	3 months	40K€	NA	NA	NA	NA	NA	Probes (IDS, IPS, scripts, etc.)	Assurance metrics	NA	NA	NA	NA	NA
ISO SAE 21434	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
CC	3	1y	87K	TOE	N/A	82	N/A	41K	Testing bench	CC input production	87	N/A	0	Testing bench	N/A
CARSEM	3	5m	51K	TOE	N/A	40	N/A	20K	Testing bench	CC input production	36	N/A	0	Testing bench	N/A
SAF	3	5m	39K	TOE	N/A	30	N/A	15K	Testing bench	CC input production	28	N/A	0	Testing bench	N/A

	Best value obtained by a framework
	Worst value obtained by a framework
	Not comparable values

Table 25 Final comparison table: cost

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

6 Reference

ANSSI. 2014. CSPN: Certification de sécurité de premier niveau des produits des technologies de l'information, réf. ANSSI-CSPN-CER-P-01, version 1.1. 7 April 2014.

Ari Takanen, , Jared D. Demott,, Charles Miller, Atte Kettunen. 2018. Fuzzing for Software Security Testing and Quality Assurance. 2018.

CARSEM: A Cooperative Autonomous Road-vehicles Security Evaluation. **Sammy Haddad, Antoine Boulanger, Pierpaolo Cincilla, Brigitte Lonc. 17-21 September 2018.** Copenhagen, Denmark, : s.n., 17-21 September 2018. 25th ITS World Congress.

Clark, Kevin, Jerald Dawkins, and John Hale. 2005. Security Risk Metrics: Fusing Enterprise Objectives and Vulnerabilities. *Proceedings of the IEEE Workshop on Information Assurance and Security.* June 2005.

Current trends and advances in information assurance metricsFredericton. **S. Nabil, P. Peter, M. Ashraf, L. Biswajit, Nandyand John, and H. Adam. 2004.** 2004, Proceedings of Second Annual Conference on Privacy, Security, and Trust (PST 2004), NB, Canada, p. 1315.

defense, US Department of. l985. Trusted Computer System Evaluation Criteria (TCSEC), DoD 5200.28-STD. 26 December l985.

ETSI. 2018. GS ISI 003 - Information Security Indicators (ISI); Key Performance Security Indicators (KPSI) to evaluate the maturity of security event detection. January 2018.

Freiling, I. 2008. *Dependability Metrics.* s.l. : Springer, vol. 4909, 2008.

ISO. ISO 26262 Road vehicles — Functional safety. s.l. : ISO.

ISO/IEC. 2009. 15408-1 Information technology – Security techniques – Evaluation criteria for IT security - Part 1: Introduction and general model. December 2009.

—. 2008. *15408-3 Information technology – Security techniques – Evaluation criteria for IT security - Part 3: Security assurance requirements.* 2008.

—. 2008. 18045 Information technology -- Security techniques -- Methodology for IT security evaluation. 2008.

—. 27004 : Information technology – Security techniques – Information security management – Measurement.

—. 27005: Information technology – Security techniques – Information security risk management.

Jaquith, A. 2007. *Security metrics, replacing fear, uncertainty, and doubt.* MA : Addison-Wesley Reading, 2007.

Jianxin Li, Bo Li, Tianyu Wo, Chunming Hu, Jinpeng Huai, Lu Liu, K.P.Lam. 2012. CyberGuarder: A virtualization security assurance architecture for green cloud computing. s.l. : Future Generation Computer Systems Volume 28, Issue 2, February 2012.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

M. Howard, J. Pincus, and J. Wing. 2005. Measuring relative attack surfaces. *Computer Security in the 21st Century Eds. Springer US.* 2005, pp. pp. 109–137.

Manadhata, Pratyusa K. and Jeannette Wing. 2006. An Attack Surface Metric. *Proceedings of the USENIX Security Workshop on Security Metrics (MetriCon).* Vancouver : s.n., August 2006.

Measuring Cyber Security and Information Assurance: a State-of-the-Art Report. **N. Bartol, B. Bates, K. Goertzel, and T. Winograd. 2009.** 2009, Information Assurance Technology Analysis Center (IATAC).

members, Common Criteria Recognition Arrangement (CCRA). 2017. *Common Criteria for Information Technology Security Evaluation v3.1 r5.* April 2017.

NIST. 2002. FIPS PUB 140-2 - Security Requirements for Cryptographique Modules . 3 Decembre 2002.

—. 2007. SAMATE. *IATAC Software Security Assurance: A State-of-the-Art Report (SOAR).* 2007.

On the brittleness of software and the infeasibility of security metrics. **Bellovin,] S. 2006.** s.l. : IEEE Security & Privacy, 2006. p. 96.

Payne, S. 2001. *A guide to security metrics.* s.l. : SANS institute, 2001.

Project, OpenSSL. 2006. Random number generator, uninitialised data and valgrind. *https://marc.info/?l=openssl-dev&m=114651085826293&w=2.* 2006.

Quality of protection: measuring the unmeasurable? . **McHugh, J. 2006.** New York, NY, USA, : ACM , 2006. Proceedings of the 2nd ACM workshop on Quality of protection. pp. pp. 1–2.

R. Vaughn, R. Henning, and A. Siraj. 2003. Information assurance measures and metrics-state of practice and proposed taxonomy. *Proceedings of Hawaii International Conference on System Sciences, vol. 1., Citeseer.* 2003.

Samuel Paul Kaluvuri, Hristo Koshutanski, Francesco Di Cerbo, Antonio Ma±a. 2013 . Security Assurance of Services through Digital Security Certificates. s.l. : IEEE 20th International Conference on Web Services, July 2013 .

Schneier, B. 1999. Attack Trees. s.l. : Dr. Dobb's Journal, December 1999.

SOG-IS. ITSEC: Information Technology Security Evaluation Criteria. *ISBN 92-826-3004-8.*

Stevens, S. S. 7 June 1946. On the Theory of Scales of Measurement. s.l. : (7 June 1946). Science. 103 (2684): 677–680. Bibcode:1946Sci...103..677S. doi:10.1126/science.103.2684.677. PMID 17750512., 7 June 1946.

TrustCB. 2018. Security Evaluation Standard for IoT Platforms (SESIP). 23 December 2018.

wikipedia. Level of measurement. *https://en.wikipedia.org/wiki/Level_of_measurement.*

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319

Willke, James F. Steven and Bradford. 2005. Enterprise Security Metrics: Taking a Measure of What Matters. *Presented at Third Annual Information Technology and Network Security Conference (SecureIT).* San Diego, California : Carnegie Mellon University Software , April 2005.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 732319