Proposal No. 830929
Call H2020-SU-ICT-03-2018

Project start: 1 February 2019
Project duration: 42 months

Cyber Security for Europe

D3.22

Validation and Certification Methodology

Document Identification	
Due date	30 April 2022
Submission date	29 April 2022
Revision	1.0

Related WP	WP3	Dissemination Level	PU
Lead Participant	CYBER	Lead Author	Liina Kamm (CYBER)
Contributing Beneficiaries	UMU, IRIT-UPS, CONCEPT	Related Deliverables	D3.8, D7.7

Abstract: We present SURFACE (Support Framework for Certification) – an integrated approach that can be used to carry out certification and recertification. SURFACE combines solutions from the Common Criteria based European candidate cybersecurity certification scheme (EUCC), the ECSO meta-scheme, the ARMOUR methodology and NIST SP 800-137. SURFACE offers support for incremental certification and uses the EUCC guidelines throughout the process.

This document is issued within the CyberSec4Europe project. This project has received funding from the European Union's Horizon 2020 Programme under grant agreement no. 830929.

Executive Summary

Based on the state of the art and the security certification challenges analysed in D7.7 (*The Role of Certification and its Implementations*), this deliverable presents a theoretical security certification framework, SURFACE (Support Framework for Certification). It is intended to manage the security level of an ICT component throughout its lifecycle thus coping with one of the mayor challenges detected, the security dynamism.

The framework combines well know efforts in the cybersecurity and certification fields:

1) The ECSO meta-scheme to allow the interoperability of different standards and certification schemes,
2) the ARMOUR methodology, which copes with the subjectivity, time and effort of the evaluation by combining risk assessment and testing in a symbiotic way,
3) the Manufacturer Usage Description (MUD) file to provide configuration recommendations to the security flaws encountered, linking the design and the runtime phases,
4) the NIST SP 800-137 to manage cybersecurity changes during the device lifecycle due to newly discovered threats or product updates, and
5) the threat MUD file to share information about compromised services and security mitigations during the operation time. All the certification process considered in SURFACE is also guided by the EUCC (Common Criteria based European candidate cybersecurity certification scheme), in accordance with the Cybersecurity Act (CSA) regulation.

The methodology proposed in the SURFACE framework will be applied to the Estonian ID card chip in D7.7 to provide a guideline on how to perform each certification step and to validate the framework.

Document information

Contributors

Name	Partner
Liina Kamm	CYBER
Jayavarshini Thirumalai	CYBER
Sara Nieves Matheu García	UMU
Petr Švenda	BRNO
Brahim Hamid	IRIT-UPS

Reviewers

Name	Partner
Daniele Canavese	POLITO
Welderufael B. Tesfay	GUF

History

Version	Date	Authors	Comment
0.1	2021-06-16	Liina Kamm, Sara Nieves Matheu García	ToC, MUD text, integrated model text
0.2	2022-01-18	Sara Nieves Matheu García, Petr Švenda, Liina Kamm	ToC pre-final, MUD text updates.
0.3	2022-01-21	Liina Kamm	ToC finalised, added section for asset description
0.4	2022-02-28	Liina Kamm, Brahim Hamid	Security assurance cases added, section 3 feature complete
0.5	2022-03-07	Sara Nieves Matheu García, Petr Švenda, Brahim Hamid, Liina Kamm	Draft edited for high level review, added asset description
0.6	2022-03-14	Liina Kamm	Technical editing (template, consistency)
0.7	2022-03-17	Sara Nieves Matheu García, Liina Kamm	Incorporated feedback from the high level review
0.8	2022-04-12	Liina Kamm	Incorporated the feedback from the first review round
1.0	2022-04-29	Liina Kamm	Added the GitHub link
1.0	2022-04-29	Ahad Niknia	Final check, preparation and submission

Table of Contents

List of Figures

List of Tables

List of Acronyms

A	**ACL**	Access control list
C	**CC**	Common Criteria
	CM	Continuous monitoring
	COTI	Challenges of the industry
	CVE	Common Vulnerabilities and Exposures
	CVSS	Common vulnerability scoring system
D	**DGC**	Dependency-guarantee contract
	DGR	Dependency-guarantee relationship
E	**EAL**	Evaluation assurance level
	ECSO	European Cyber Security Organization
	EG	Expert group
	ENISA	European Union Agency for Cybersecurity
	ETR	Evaluation technical report
	ETSI	The European Telecommunications Standards Institute
	EUCC	European Candidate Cybersecurity Certification Scheme
F	**FIPS**	Federal Information Processing System
G	**GDPR**	General Data Protection Regulation
	GPP	Generalised protection profile
	GSN	Goal structuring notation
	GST	Generalised security target

I	**ICT**	Information and communications technology
	IEC	International Electrotechnical Commission
	IETF	Internet Engineering Task Force
	IoT	Internet of things
	ISO	International Organization for Standardization
	ITSEF	IT Security Evaluation Facilities
M	**MBT**	Model-based testing
	MRA	Mutual recognition agreement
	MUD	Manufacturer usage description
N	**NIST**	National Institute of Standards and Technology
O	**OCL**	Object Constraint Language
R	**ROE**	Rules of engagement
S	**SA**	Safety argument
	SC	Safety case
	SURFACE	Support framework for certification
	SUT	System under test
T	**TOE**	Target of evaluation

1 Introduction

The advent of paradigms such as the 5G technology or the Internet of Things (IoT) promises to realise the vision of a hyper-connected society, in which humans and devices compose complex interconnected systems leading to a strong cybersecurity interdependence. In this scenario, the final network becomes much more complex and therefore it can be much more feasible for a vulnerability to affect many more systems and to be propagated very quickly. The borderless nature of the infrastructures and threats involved also means that any vulnerability or security incident in one country can have disastrous consequences in the whole European Union.

While Europe is leading large initiatives to guarantee the security of these systems, such as the Cybersecurity Act[1] or the 5G toolbox[2], it is still not yet clear how to deal with vulnerability dependencies in such a complex environment.

Certification of products and services helps to systematically test and assess the security targets, and the certificate provides more assurance to the consumers. However, there are very many different standards and protection profiles out there to choose from and the results are often not easy to comprehend by someone not involved in the process. Moreover, the assurance reports and resulting certificates present the information as free text, and are, therefore, difficult to process automatically. The dependencies to other products and certificates are static, and when something happens to the connected certificate, the issue is often not propagated.

We present SURFACE (Support Framework for Certification) – an integrated approach for the process of certifying and recertifying. SURFACE brings together solutions from EUCC (Common Criteria based European candidate cybersecurity certification scheme) [1], the ECSO meta-scheme [2], the ARMOUR methodology [3] and NIST SP 800-137 [4]. We have combined the solutions in such a way that they complement each other at different steps. The ECSO meta-scheme allows the integration of certification schemes or standards. The ARMOUR methodology supports SURFACE in establishing the context, testing and communicating the result processes. NIST SP 800-137 supports continuous monitoring for patches or updates. SURFACE supports incremental certification, which reduces the cost and time taken for recertification. SURFACE uses the EUCC guidelines throughout the process starting from the selection of assets to the recertification process. Hence, it is in accordance with articles of the Cybersecurity Act.

SURFACE also takes advantage of the cybersecurity certificates information and the MUD files to manage security dependencies and provide mitigations. This way the dependencies can be traced when a new threat is discovered. On the one hand, the certificate indicates certified subcomponents that the system has, and on the other hand, the MUD file indicates the connections with other services not certified or not considered in the certificate. Moreover, knowing the affected services, we can apply fast mitigations before a patch or update is released by the manufacturer.

A structured certification report makes the result more easily analysable and helps discover dependencies between certificates. If a new vulnerability occurs and a certificate is revoked, other affected certificate holders can automatically be notified.

The SURFACE approach will be validated in D7.7, using the Estonian ID card chip as an example.

[1] https://eur-lex.europa.eu/legal-content/EN/ALL/?uri=CELEX%3A32019R0881
[2] https://ec.europa.eu/commission/presscorner/detail/en/qanda_20_127

1.1 Structure of the Document

In Section 2 we give an overview of the building blocks used in creating the integrated theoretical framework. Some of the elements will be described in more detail in Deliverable 7.7, in this deliverable we give a very short summary of these for easier reference. Section 3 describes SURFACE and its certification steps. Subsection 3.10 highlights the innovation of our solution over the state of the art. Section 4 proposes a method to derive argument patterns to construct security assurance cases from formal methods in Alloy. Section 5 describes updates to the certification assistant tool that was introduced in Deliverable 3.8, and which has been redesigned based on SURFACE. Section 6 gives the conclusion and can used as input for GitHub, where all the asset descriptions are kept for a concise overview.

2 Building Blocks

2.1 ARMOUR Methodology

The ARMOUR methodology [3] is developed using one of the ETSI proposals [5]. It integrates ISO 31000 [6], extended control assessment and the testing actions defined in ISO/IEC/IEEE 29119-1 [7]. All the processes involved in the ARMOUR methodology are represented in Figure 1. The two major underlying themes are security testing and security risk assessment where the results of one process can be used to improve the other. Both are initiated by establishing the context. The results of these activities can be managed for any changes or updates and shared by the processes called monitoring and review and communicate and consult.

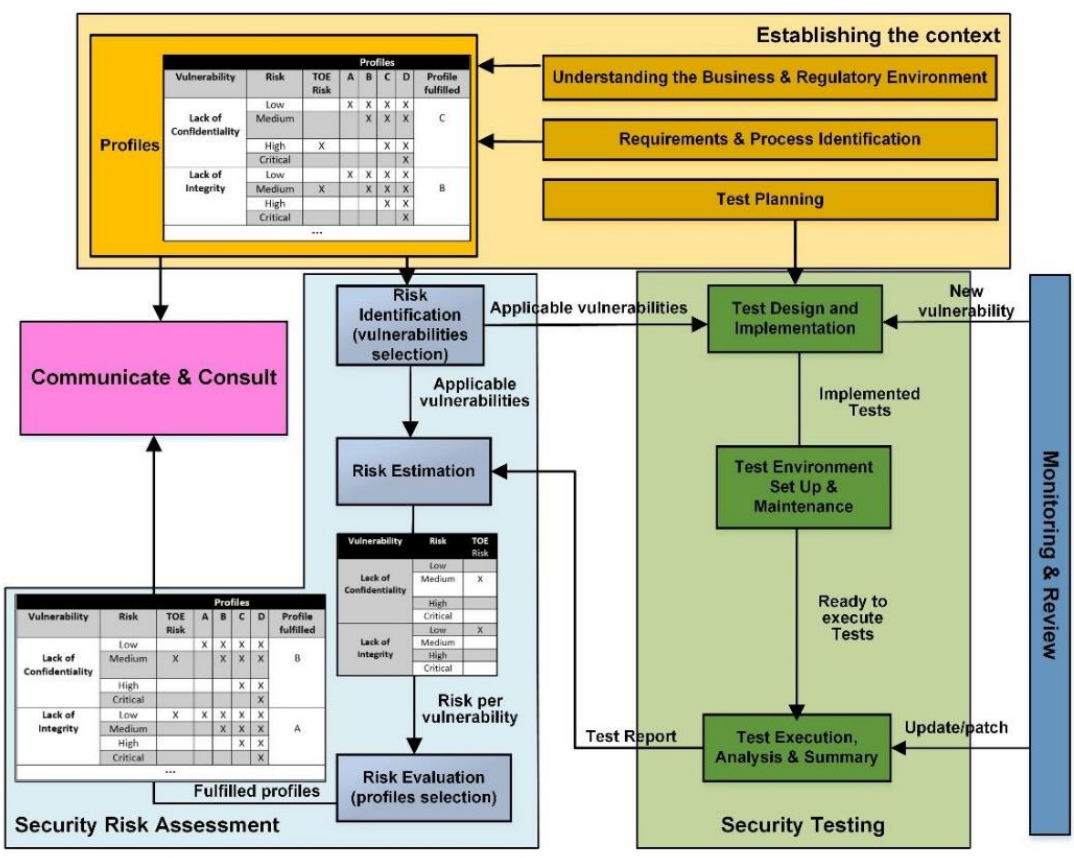


Figure 1: ARMOUR methodology processes

2.2 Manufacturer Usage Description (MUD) Standard

The manufacturer usage description (MUD)[3] was standardised in 2019 within the scope of the Internet Engineering Task Force (IETF). The MUD specification's major goal is to limit the threat and attack surface of a certain IoT device by allowing manufacturers to establish network behaviour profiles for their devices. Each profile is built around a set of policies, or access control lists (ACLs), that specify the communication endpoints. MUD represents a scalable and flexible approach to the definition of network access policies beyond the use of IP addresses to enable communications with other services. A manufacturer could, for example, declare that access to particular cloud services, as well as connection with other manufacturers' devices, should be permitted. MUD also allows specifying protocols and ports for each communication to provide a more fine-grained configuration of access control rules. The standard also makes it possible to extend the scheme, allowing manufacturers to express other types of conditions or policies based on their needs. For example, while the MUD is focused on network access control regulations, MUD model expansions are being considered for quality of service aspects of the communications. However, despite this flexibility, the MUD model does not provide mechanisms to describe more fine-grained aspects and additional security restrictions beyond the network layer. Dealing with this limitation, authors in [8] [9] propose an extended version of the standard.

One of the key advantages of the MUD approach is that the manufacturer is responsible for defining the devices' behavioural profiles (instead of the typical network administrator). Indeed, the MUD design and format make it possible to automate the creation of network access policies based on the manufacturer's MUD profile. It should be noted, however, that the instantiation of these profiles may be influenced by the network domain in which the device is deployed. The standard also defines an architecture to allow the network domain where the device is deployed to obtain and enforce this profile. In particular, the MUD architecture is composed by four main components:

- Thing or Device, which is in charge of sending a MUD URL to indicate where its MUD file is stored.
- Router or Switch to which the device is connected.
- MUD Manager, the main entity to manage MUD files, which is in charge of the processes required for obtaining and enforcing the different security restrictions described in a MUD file.
- MUD File Server, which hosts the MUD files of a particular manufacturer.

The MUD standard restricts IoT device connections by defining ACLs, using the Yet Another Next Generation (YANG)[4] standard to model network restrictions and using JavaScript Object Notation (JSON)[5] for serialisation. It is worth noting that the MUD model includes Network ACL extensions to the YANG data model, which are augmented by the MUD standard to specify more expressive ACLs.

The MUD file contains two main blocks: the “mud” and “acls” containers. Whereas the first block specifies several features related to the MUD file itself e.g., last update, MUD URL, version, manufacturer, the second block (Figure 2) defines the ACLs, specifying the conditions and actions to apply in case they are fulfilled (e.g., allow or deny).

[3] https://datatracker.ietf.org/doc/html/rfc8520
[4] https://datatracker.ietf.org/doc/html/rfc7950
[5] https://datatracker.ietf.org/doc/html/rfc7159

```
module: ietf-mud
     +--rw mud!
        +--rw mud-version           uint8
        +--rw mud-url               inet:uri
        +--rw last-update           yang:date-and-time
        +--rw mud-signature?        inet:uri
        +--rw cache-validity?       uint8
        +--rw is-supported          boolean
        +--rw systeminfo?           string
        +--rw mfg-name?             string
        +--rw model-name?           string
        +--rw firmware-rev?         string
        +--rw software-rev?         string
        +--rw documentation?        inet:uri
        +--rw extensions*           string
        +--rw from-device-policy
        |  +--rw acls
        |     +--rw access-list* [name]
        |        +--rw name    -> /acl:acls/acl/name
        +--rw to-device-policy
           +--rw acls
              +--rw access-list* [name]
                 +--rw name    -> /acl:acls/acl/name
```

Figure 2: ACL definition in a MUD file

Since its adoption, MUD has received significant interest from the research community and standardisation bodies. In particular, the National Institute of Standards and Technology (NIST) proposes the MUD standard as a promising approach to mitigate security threats, and to cope with denial-of-service attacks in IoT environments, including home and small-business networks [10]. Additionally, the European Union Agency for Cybersecurity (ENISA) considers the use of MUD as part of IoT security good practices to improve, allowing devices to advertise their supported and intended functionality.

2.3 European Cybersecurity Candidate Scheme

The European cybersecurity candidate scheme (EUCC) [1] from ENISA provides a set of guidelines, rules and regulations based on Common Criteria for an ICT product evaluation. These guidelines are in compliance with the requirements of the Cybersecurity Act. The major benefits of implementing standardised criteria and methods for cybersecurity evaluation are, for example, consistency between different manufacturers and vendors, reusability, harmonisation of terminology, and awareness.

2.4 ECSO Meta-scheme Approach

The major goal of ECSO is to provide standardised cybersecurity solutions for supporting and protecting digital processes. One of the results from ECSO work on standardisation, certification and supply chain management (WG1) is the meta-scheme approach [2]. The meta-scheme approach focuses on two concepts: security and privacy. We consider two WG1 documents, Challenges of the Industry (COTI) and the State of the Art Syllabus (SOTA) [11] to help us in addressing drawbacks in existing schemes and standards.

The ECSO meta-scheme involves the integration of various certification schemes and standards for the conformity of items. This integration can be achieved by defining a high-level common language or format across the schemes selected for evaluation. The expert group (EG) is responsible for defining the common language where the EG represents security researchers or specialists. For evaluating an item, it is important to understand the scope of the targets from which the EG will be drawing related threats and making assumptions. Thus, some modified notions are cascaded from Common Criteria. Initially, a Generalised

Protection Profile (GPP) is defined in the common language for the security target. A GPP is composed of the following terms [2]:

- the security problem definition that deals with defining the assets, their threats, assumptions, constraints,
- the security objectives that deal with identifying the security requirements,
- the security services and features related to the functionalities of the target,
- the instantiation of the levels (selection of schemes from SOTA),
- the visual representation of the minimum required scope of security functionality per level using the cybersecurity label.

The developed GPP has to be approved by an accredited third party. Similar to GPP, the meta-scheme uses the term Generalised Security Target (GST) to represent the target of evaluation (ToE). The assurance level is generated based on the type of penetration testing carried out (black-box, white-box and grey-box) on the target. The relation between the scope of the security functionality and the assurance level helps the EG to appropriately select and modify the schemes available from SOTA for the identified issues.

The ECSO SOTA is a publicly available dynamic document that consists of all the available certification schemes used for security certification and also standards that are related to cybersecurity for various assessments. This document helps in providing certification to a component or a part of the component, products and organisations. For every scheme or standard and specification, SOTA provides brief descriptions for the focus, applicable area, associated scheme and governance, process, practice and relation to other standards or schemes. The certification schemes or standards provide a set of regulations or guidelines. Based on these the conformity of a product or any system is approved or denied. The certification schemes are required to be generic rather than sector dependent, since the technologies that are used in developing a solution may be used in multiple domains.

2.5 Security Control Assessments

NIST SP 800-53A [12] defines three different assessment methods: examine, interview and test. Of these, examining is more time consuming. The interview method requires manual work to achieve good results. The test method is the most effective for automation and provides more accurate results with proper implementations [13]. When the assessment method is chosen, assessment types need to be decided. The following subsections give a short overview of the different types of assessments and the features based on EUCC and the ECSO meta-scheme.

2.5.1 ECSO Assessment Options

Some organisations carry out self-attestation which is either a declaration without any assessments, or declarations based on a self-assessment or third-party assessment. A product or a system can be assessed in three different ways [14].

Self-assessment. Generally, self-assessment is not considered an effective assessment approach. As defined in the Cybersecurity Act (CSA) article 54.1, we need to verify whether self-assessment is accepted by the schemes. For instance, self-assessment is allowed only in the basic assurance level and this level is not accepted by the EUCC scheme. It is hard to predict whether the self-assessment was accredited, as it is carried out by the organisation itself.

Third-party assessment. Unlike self-assessment, third-party assessment is carried out by an accredited third-party. It can be carried out by either an in-house body or an external body. The in-house assessment is

led by the organisation and the involved activities are validated by the National Accreditation Body or an accredited third-party. External assessment is carried out by an accredited third-party and the activities are inspected by the national accreditation body.

National third-party assessment. This type of assessment is generally carried out by the national entity for ensuring national security for example in the case of military and nuclear projects.

2.5.2 Testing Approaches

The fundamental goal of security testing is to ensure that the software meets the requirements for major security properties such as confidentiality, integrity, availability, authentication, authorisation and non-repudiation. The system under test (SUT) can be a system, service, application or any other digital item which is being tested with the baseline security controls. To select the appropriate testing techniques, we have analysed the features, pros and cons of various testing methods from [15]. The following two approaches have a significant role in our theoretical framework.

Model-Based Testing. Model-based testing (MBT) [15] requires us to design and create an efficient model that represents the SUT, its environment and its behaviour. One of the main advantages of MBT is that we can partially automate the recertification process. MBT is carried out by designing the models using a high-level representative language such as Object Constraint Language (OCL) or Unified Modelling Language (UML). The test model is designed manually. The test cases can be generated automatically based on the model [16]. There are various tools such as CertifyIt and MISTA for generating the tests [17].

Penetration Testing. Penetration testing [18] can be used to simulate real-time attacks for detecting vulnerabilities and exploiting them. The testing can be black-box, white-box or grey-box. All of these variations require high technical knowledge and skill for carrying them out. Black-box penetration testing starts from the information gathering process on the target and simulates an outsider attack. White-box penetration testing provides the testers with additional information about the target, e.g., network architecture, source code for carrying out the test. In grey-box penetration testing, the tester gets partial knowledge about the target, e.g., administrator account credentials, and simulates a combination of internal and external testing.

In addition, other testing approaches [15] may be carried out as part of penetration testing based on the SUT requirements. For instance, experts may carry out source code analysis or fuzzing during a web application penetration testing to improve code quality. Some examples of documentation or manual for carrying out penetration testing like Open Source Security Testing Methodology Manual (OSSTMM) [19], Penetration Testing Methodologies and Standards (PTES) [20], Open Web Application Security Project (OWASP)[6] testing guide. The testing methods can be combined, e.g., MBT can be combined with fuzzing to provide a more efficient method.

2.5.3 Risk Assessment

Risk assessment process helps an organisation to create better business continuity and disaster recovery plans by analysing the threats associated with valuable assets. Risk assessment can be quantitative, semi-quantitative and qualitative [21]. The organisation can choose the appropriate assessment type based on its business process or environment. There are various risk assessment approaches such as the Common

[6] https://owasp.org/www-project-web-security-testing-guide/v42/

Weakness Scoring System (CWSS)[7], the Veracode Rating System[8], the OWASP risk rating methodology[9] and the Common Vulnerability Scoring System (CVSS)[10].

The CVSS is an open-source framework that provides three groups of metrics: base, temporal and environmental. Initially, the base metric is calculated based on the exploitability metric and the impact metric. This base metric is represented as a numerical value in the range of 0.0 and 10.0, but this can be changed based on the temporal and environmental metric scores. CVSS allows the representation of these scores through text (low, medium and high). CVSS v3.1 is the latest version in which the functionality of metrics is modified to some extent. The CVSS is used in various platforms including Common Vulnerabilities and Exposures (CVE) created by the MITRE corporation[11], and the NIST Vulnerability Database.

2.6 Additional Elements

2.6.1 Protection Profiles

Protection Profiles (PPs) are developed to represent the different contexts or scenarios of the security target. The EUCC and CC prefer the PPs to be certified first before using them to certify a product. In addition, the EUCC requires each certified PP certificate to contain information like a unique certificate ID for the PP and technical and non-technical details of the PP [1].

In this document, the term GPP is used for representing integrated PPs. The EG can either make use of the existing PPs or a GPP is developed based on the chosen certification schemas or standards or other related PPs. The developed GPP is then certified by the accredited testing laboratory (ITSEF) to verify whether they are in compliance with the CC requirements.

2.6.2 Cybersecurity Label

Lack of transparency is one of the issues mentioned by ENISA to consider during the security certification of any ICT product [22]. When a product is certified successfully, it is possible to create a label, which is valid for a specific period of time. Labelling increases the transparency of the security assurance of the product. Every technical and non-technical person should be able to visually identify the difference between a validated and invalid product. It is not possible for every person to check the security strength of an item where the evaluation involves a series of steps. To overcome this issue, the cybersecurity label can be used to indicate a properly validated product.

Finland was the first country in Europe to initiate the cybersecurity label for smart devices based on EN 303 645[12]. ECSO has introduced a label called Cybersecurity Made in Europe[13] to indicate that the application or product has been developed by a European company or organisation. Their intention for issuing this label is to broaden the European market by verifying the geo-location of the target company and

[7] https: //cwe.mitre.org/cwss/cwss_v1.0.1.html.
[8] https://help.veracode.com/r/DGHxSJy3Gn3gtuSIN2jkRQ/civ7DGQfn2Kk4xh4Cz4UtA
[9] https://owasp.org/www-community/OWASPRiskRatingMethodology
[10] https://www.first.org/cvss/v3-1/cvss-v31-specification_r1.pdf
[11] https://www.mitre.org/
[12]https://www.traficom.fi/en/news/finland-becomes-first-european-country-certify-safe-smart-devices-new-cybersecurity-label
[13] https://ecs-org.eu/working-groups/cybersecurity-made-in-europe-label

not for assessing the security intensity of a product. In the ECSO meta-scheme approach, the cybersecurity label is represented using the radar diagram and in the ARMOUR methodology, QR codes based on the scenarios are used to represent the security strength.

2.6.3 Assurance Level

In general, the process of assurance is a way to ensure that the product is in compliance with the baseline security requirements through evaluation and the identified vulnerabilities are patched. All certification based on the CC certification scheme ensures the assurance level through the evaluation assurance level (from EAL1 to EAL7) based on the level at which the system was tested. Similarly, the ECSO meta-scheme and the EUCC define that the assurance level can be declared based on control assessment [1] [2]. For instance, in the ECSO meta-scheme, the level is represented as base (entry, basic) and advanced (enhanced basic, moderate and high) based on the type of penetration testing and assessment body.

2.6.4 Monitoring

Monitoring is defined as gathering and analysing the information with the help of continuous monitoring to make a decision in case of exceptions or incidents [13]. The monitoring of the product must ensure that certificates and products are valid to use. The process ensures that every component in the system meets the security requirements all the time. In SURFACE, continuous monitoring is defined as in NIST SP 800-137 (Information Security Continuous Monitoring) [4] in compliance with the EUCC rules.

2.6.5 Recertification

Recertification can be either periodic or based on necessity when system modifications occur. With continuous monitoring, we can detect if a change or update has been carried out on the target. If this is the case, it is necessary to repeat the certification process for either the whole system or a particular component depending on the change that has been made. For instance, there are numerous certifications to ensure the knowledge and competence of a person. These certificates expire after a certain amount of time. To ensure the candidate is aware of the latest technologies and processes, he or she has to periodically retake the certification test to prove their knowledge. Similarly, a certificate issued for a product is valid for a certain amount of time and has to be updated periodically. Recertification can be triggered for the following reasons or situations:

- a feature is newly added,
- there is an update or modification in a feature,
- there was an attack on the target,
- a zero-day vulnerability has been found,
- the certificate has expired,
- when there is a major change in the requirements of the target.

We define recertification based on the modular and incremental certification approach [23] developed by the Industrial Avionics Working Group (IAWG, an industrial consortium). If the listed situations affect the functionalities of the product, then a complete recertification of the product is required. Otherwise, the specified components can be recertified based on the concept of incremental certification.

3 SURFACE - Support Framework for Certification

3.1 Certification and Recertification Process

The certification and recertification process of SURFACE (support framework for certification) is described on a high level in Figure 3. The processes are divided into phases inspired by the ARMOUR methodology:

- Phase 0 (reconnaissance) where we identify the scope and select the appropriate certification schemes;
- Phase 1 (planning), where we establish the context, develop a Generalised Protection Profile and plan the security control assessment process;
- Phase 2 (assessment), where we carry out the security testing and risk assessment;
- Phase 3 (generating certification elements), where we generate the certificate and the accompanying materials;
- Phase 4 (communicating the results), where we share the certification elements with experts and clients;
- Phase 5 (recertification), where we decide whether to recertify the whole target or components; and finally,
- the continuous monitoring process.

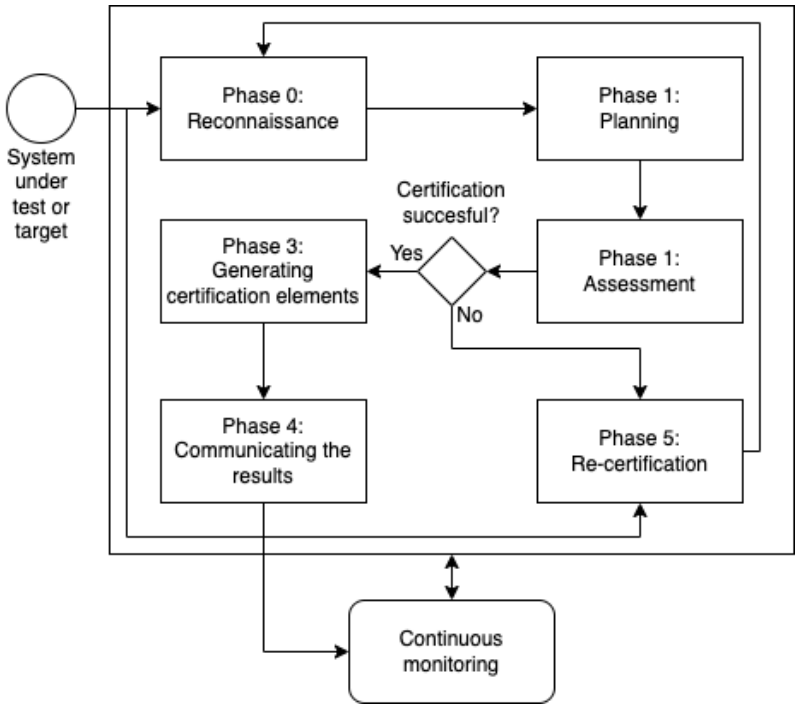


Figure 3: High level overview of SURFACE

A more detailed overview can be seen on Figure 4. The SURFACE framework integrates and combines steps and processes from the ARMOUR methodology, the ECSO meta-scheme, the European Cybersecurity Candidate Scheme and the continuous monitoring process from NIST 800-137. Our process starts with establishing the context, next we move to security assessment, and then to the communicate and consult processes. SURFACE combines the communicate and consult process as communicating the results.

We found that each of the solutions is missing some processes. The ECSO meta-scheme approach is more general than the ARMOUR methodology and allows the certification of different products and services (not only IoT devices). However, it lacks processes for recertification or continuous monitoring. Another drawback of the meta-scheme approach is that there is no common platform or database to share the findings and results. Combining solutions from different approaches can help in creating a solution that deals with the shortcomings.

By following the guidelines from the EUCC, we can define a standardised framework that supports multiple ICT products. From the ECSO meta-scheme, we use a common language (Generalised Protection Profiles, GPP) to select and integrate the appropriate certification schemes or standards. We also use the term expert group (EG) to represent the security researchers connected to the process.

SURFACE does not accept self-assessment as a trustworthy evaluation. We define the security assessments as processes that are carried out by a conformity assessment body or an accredited third-party [1]. The testing laboratory is referred to as IT security evaluation facilities (ITSEF). As a result of an assessment, it is possible to evaluate whether the security objectives, assurance level and selection of security schemes are in accordance with the CSA Articles 51, 52 and 54 respectively.

For risk assessment, SURFACE uses semi-quantitative risk assessment. To support risk communication, the generated risk scores (quantitative) can be mapped to the severity levels (qualitative). As the default, we use CVSS v3.1 to generate the severity levels of the exploitable vulnerabilities.

SURFACE uses a cybersecurity label similar to the ARMOUR methodology. This helps to represent the certified product as well as the security level of the security properties. The framework allows the vendors to decide whether to print the label on the certified target.

SURFACE represents the assurance level as basic, substantial or high based on the type of penetration testing carried out on the target. The assessment body should be an accredited third-party.

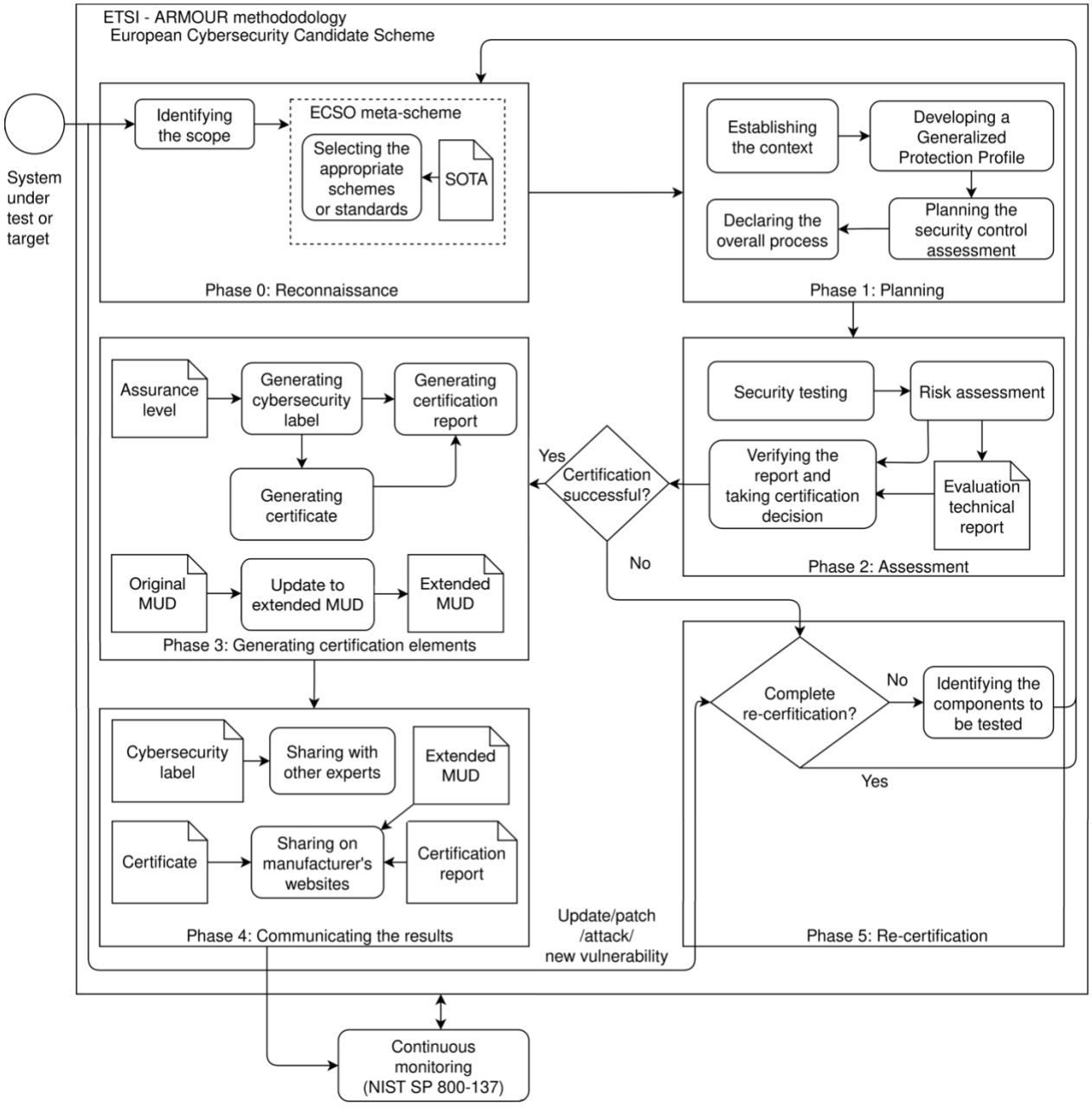


Figure 4: Detailed overview of SURFACE

3.2 Phase 0: Reconnaissance

The certification schemes or standards provide a set of regulations or guidelines. Based on these the conformity of a product or any system is approved or denied. The certification schemes are required to be generic rather than sector dependent, since the technologies that are used in developing a solution may be used in multiple domains. The first step is to decide which certification schemes or standards to use for certifying the product.

The idea of our certification framework is to combine the requirements of the chosen schemas and standards using the Generalised Protection Profile (GPP). We integrate based on the common language from the ECSO meta-scheme. The framework provides a higher level of assurance than using a single evaluation scheme or standard.

In this phase, the expert group (EG) gathers the necessary technical and non-technical information about the system under test (SUT). Information gathering is required for screening the certification schemes or standards and threats that are associated with the SUT. Once the EG has sufficient knowledge about the SUT, they can proceed. The EG will select threats and map them to security properties or vulnerabilities of the SUT. The vulnerabilities help create a Generalised Protection Profile (GPP) and are used in the testing process to verify the conformity of the SUT.

Using SURFACE, the EG can integrate the schemes based on common language from the ECSO meta-scheme. This is done during the planning phase (Phase 1). The EG is also responsible for generating the rules of engagement (ROE) based on the template from NIST SP 800-115. It contains information like the point of contact, constraints and scope. This ROE is developed for the IT Security Evaluation Facilities (ITSEF), who has to follow these rules during the assessment. In addition, the EG declares the privacy policies that are to be maintained by ITSEF.

3.3 Phase 1: Planning

3.3.1 Establishing the Context

In the planning phase, the EG establishes the context defining the scope and purpose of the target. The EG analyses the SUT requirements and functionalities in terms of appropriate security properties, business processes, working environment and related laws. As a result of context establishment, the EG derives different profiles with unique names. These profiles represent different features of the SUT along with acceptable risk levels specific to appropriate vulnerabilities. The acceptable risk levels are generated based on the business and environmental conditions. Also, the acceptable risk levels will be compared with the actual risk levels in the assessment phase (Phase 2) to make a decision (Section 3.4.3). With the help of context establishment, the EG can also develop safety cases required for incremental recertification (discussed in Section 3.7).

In SURFACE, the SUT is allowed to certify either against a specific Protection Profile (PP) or the EG can create its own GPP based on the results of the reconnaissance phase, the context establishment phase and different existing PPs. The EG also has to specify the constraints (if any) related to the SUT. Every PP or GPP is required to possess:

- the security problem definition based on threats or vulnerabilities, assets, constraints and assumptions for the target;
- the security objectives based on problem definition with respect to the considerations of EUCC;
- the security services and features based on the objectives and functionalities of the SUT; and
- the instantiation levels which contain the selected schemes, the list of created profiles, acceptable risk levels and evaluation steps.

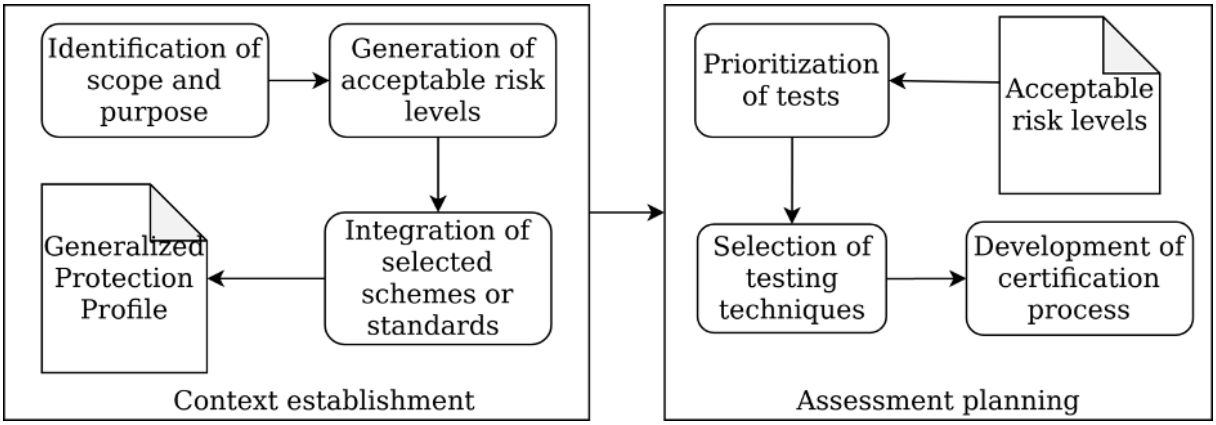


Figure 5: Phase 1: Planning

Figure 5 represents the activities involved in the planning phase. As mentioned, in EUCC and CC, the developed GPP is evaluated by an accredited ITSEF. It is sufficient to validate the GPP against the protection profile evaluation (APE) criteria and the security target against the security target evaluation (ASE) criteria from the CC (Part-3). This evaluation is done to ensure that every GPP complies with the requirements of CC as they are developed based on more than one scheme or standard.

If the evaluation result is successful, a unique ID will be provided to the evaluated GPP. If the evaluation is unsuccessful, the EG is required to carry out appropriate changes on the GPP or security target. If the vendor or manufacturer prefers, a certificate can be generated for the (positively) evaluated GPP. The format and content for this certificate should be declared based on the EUCC. The GPP is allowed to undergo any relevant updates or changes based on the system requirements. The GPP certificates can be added as a subset of the certificate for the system or product, and it is not mandatory to define the GPP under the supplementary cybersecurity information. This supplementary cybersecurity information is further discussed in Section 3.6. Though the GPPs are subject to changes, they do not necessarily require separate monitoring measures.

Based on the information gathered, a test plan is created, where testing techniques, process flow, and testing priorities for the remaining certification and recertification stages are declared. Note that acceptable risk levels (for each profile) can be reused during the periodic recertification process assuming that there have been no attacks on nor updates to the service or product. Otherwise, the process should restart from Phase 0.

3.3.2 Assessment Planning

The decision to approve or deny the certificate is based on the assessment results. Assessment is carried out using the threats gathered in Phase 0. In this phase the assessment activities are planned based on the certification scheme(s) and standard(s) that were selected by the EG in Phase 0. Risk analysis of each threat can help in determining the acceptable risk levels that are specific to each profile.

Based on the acceptable risk levels, the tests are prioritized and executed. Based on the ARMOUR methodology, the following activities are carried out in assessment planning:

1. prioritising the tests for the vulnerabilities based on the acceptable risk levels;
2. selecting the testing techniques;
3. selecting the risk assessment approach;

4. developing an overall certification plan or process.

The assurance levels of certification are decided based on the chosen testing techniques. Hence, to provide high assurance, dedicated security testing activities are required. The testing techniques are selected by the EG and it is possible to use more than one testing technique. In addition, the EG is responsible for deciding which risk assessment approach to use. Although SURFACE also allows to carry out self-assessment of conformity, it does not consider this as accredited assessment. Hence, the testing and risk assessment processes have to be carried out by an accredited third party (ITSEF).

Before beginning the evaluation, all the mandatory supporting elements and guidance supporting documents are provided. These documents help the evaluator to gain more knowledge on the functionalities of the SUT, its requirements and also guidance on carrying out the evaluation. Where applicable, the evaluator must be supplied with all the necessary and appropriate information during the evaluation process. We highly recommend exchanging information based on the privacy policies from the EUCC. The supporting documents that are used by the evaluator during the certification and recertification processes are given in the evaluation technical report (ETR) and the certification report.

3.4 Phase 2: Assessment

The assessment phase can be considered the most important step of certification or recertification, as it has a direct impact on the certification decision (Section 3.4.3). ITSEF or the testers are responsible for carrying out testing and risk analysis for the SUT. They are provided with all the mandatory documents along with the rules of engagement. The related vulnerabilities along with the acceptable risk levels of the system are provided as inputs to the testing process.

CC provides a set of responsibilities that have to be carried out. However, they do not provide the procedures or steps on how to accomplish them. In SURFACE, we have analysed various testing methods and risk assessment approaches to carry out the evaluation. These are described in more detail in Deliverable 7.7, and a short overview is also given in Section 2.5.

3.4.1 Testing

Based on the chosen testing method, ITSEF tests the system for the selected vulnerabilities. The testing process (e.g., generated model-based testing designs, penetration test plans) can be reused (directly or with minor modifications) during the recertification process. ITSEF are allowed to use any appropriate automated testing tools. In addition to testing against selected vulnerabilities, ITSEF can either manually, or using a vulnerability scanner, identify new or zero-day vulnerabilities. If any new vulnerabilities are found during the penetration testing process, a copy of that vulnerability is sent to Phase 1 so that it can be mapped to the appropriate security property according to its context or domain.

When testing is completed, ITSEF reports all the findings as a technical report. This technical report is given as input to the risk assessment process.

3.4.2 Risk Assessment

The CC process does not include risk assessment. However, to address and prioritize the vulnerabilities or threats, we have decided to include risk assessment in the SURFACE framework. The exploitable vulnerabilities (identified during testing) are given as input for risk analysis to determine the actual risk level with respect to the context or profile. Different methods can be used to determine the risk level.

The risk score is determined for every profile related to the exploitable vulnerability. The estimated score is then mapped to the intervals given by the methodology. There are some exceptional cases where the vulnerabilities are provided with the default risk level. Based on the ARMOUR methodology, vulnerabilities with no protective measures are classified as critical risk and the vulnerabilities that cannot be exploited at present are classified with the low risk level. SURFACE allows ITSEF to use any accredited risk assessment

tool (certified against any international standard) to carry out the risk assessment on the exploitable vulnerabilities where applicable. This can allow ITSEF to partially automate the estimation of risk scores. ITSEF can propose risk mitigation methods that the EG can use to prepare mitigation activities. Risk treatment is not considered to be in the scope of this deliverable and will be handled as future work.

The results of the testing and risk assessment are compiled into an evaluation technical report (ETR). The ETR must contain sufficient information about the assessment which will be validated by the certificate authorising scheme. For instance, the ETR should contain information about the type of testing used in the evaluation along with valid evidence to prove the assurance level on the SUT. The ETR can be reused or referenced during future testing. The ETR and other sensitive information of assessment must be shared only with the manufacturer of the SUT (or an approved person), as the details of exploitable vulnerabilities can pose a huge threat to the manufacturer and users of the system.

3.4.3 Certification Decision

Based on the outcome of the testing and risk assessment, the certificate authorising scheme determines whether the certification approval can proceed. The actual state of the baseline security of the SUT is compared with the expected state. The risk levels obtained from the risk assessment are compared with the acceptable risk levels generated by the EG in the planning phase. The profile comparison is based on the comparison described in Deliverable 3.8. Every profile whose acceptable risk level matches the actual risk level is considered as fulfilled.

When all the profiles are fulfilled, the SUT is considered to be eligible for certification and proceeds to the next phase. In addition to the profile fulfilment, evidence for performed evaluation (ETR) must be provided to the certificate authorising scheme [1]. The certificate authorising scheme is also responsible for making sure that the mutual recognition agreement (MRA) is not violated and the rules of engagement mentioned in the MRA are satisfied. When a profile is not fulfilled, then the identified threats are prioritised for recertification (after mitigation) based on the risk level. Table 1 represents the certification decisions based on the assessment results and the evidence provided.

Condition	Decision
The SUT meets the requirement criteria	Issue the certificate
The certificate of the SUT expired, no updates or modifications were done, no attacks were discovered, and the new assessments were successful	Continue the certificate and extend the validity
The certified SUT components had updates or modifications, or an attack was discovered (certificate may or may not have expired) and the new assessments were successful	Renew the certificate with extended validity

The certificate of the SUT expired, no updates or modifications, or an attack was discovered and the new assessments were not successful	Suspend the certificate validity. Proceed with recertification after remedial measures
The certificate of the SUT expired and the vendor did not request certificate maintenance	Archive the certificate
The certified SUT components had updates or modifications, or an attack was discovered (certificate may or may not have expired) and the new assessments were not successful	Suspend the certificate and proceed with recertification after remedial measures
The necessary assessments were not successful for the same SUT version, but works with reduced assurance level or scope	Continue or renew the certificate with reduced assurance level or scope and extend its validity
The assessments were not successful and no possible actions can be performed	Do not issue certificate or withdraw the certificate
Improper certificate or cybersecurity label usage	Suspend the certificate and the respective authority should make corrective measures
Proper remedial or corrective measures are not taken within the given time	Do not issue certificate or withdraw the certificate

Table 1: Certificate decisions

3.5 Phase 3: Generating Certification Elements

Once the certificate is issued, it should always be accompanied by the certification report. For a better visual overview, similarly to the ARMOUR methodology, SURFACE also encourages the use of a cybersecurity label.

3.5.1 Certification Report

A certification report contains all the information about the system certification process. The report is based on the evaluation technical report (ETR) by the certificate issuer. The report can be published on the CC portal or vendor's site based on the availability conditions discussed in Section 3.6. The report should follow

the content and format described in the EUCC. This leads to the creation of a unified report format for all vendors. Based on the based EUCC, the content of the certification report in SURFACE is the following:

1. executive summary, an overview of the assessment results;
2. target of evaluation details, both technical and non-technical information;
3. scope and assumptions, environmental details, limitations in evaluating the target, the threats that were considered in assessing the target;
4. security policies, rules or constraints that ITSEF should comply with;
5. certification schemes or standards, information about the schemes and standards involved in certifying the target;
6. supplementary cybersecurity information, all the information is added with respect to CSA Article 55;
7. conformity assessment body, ITSEF details;
8. evaluation results, details about testing techniques, tools used and assessment results;
9. certificate details, which contain unique ID, issuance date, validity;
10. summary, which includes the certification decision, description of the security level of the target;
11. cybersecurity label, the generated label can be added in the report;
12. bibliography, references to the supporting documents like technical documentation of the target.

If all certification reports contain this information in a structured format, it will allow for an easy overview and comparison of different certificates. A reference to the certification report can be added to the certificate. In addition, vendors can generate the European Cyber Security Certificate report [2] or any other summarised report based on the certification report, if required.

3.5.2 Cybersecurity Label

The cybersecurity label for the system should be generated only when the certification is completed successfully. It is valid only for a certain period of time (based on certification validity). If the assessment criteria are not fulfilled, the use of the cybersecurity label for the system is prohibited [1]. Based on the ARMOUR methodology, the SURFACE framework creates a label with the following components:

1. the QR code with a link to the generated certificate to guarantee its continuous update,
2. the security properties and their level present on the system,
3. the assurance level: basic, substantial or high,
4. validity information.

SURFACE allows the vendor to generate a multi-dimensional label to represent the level of different security properties. This cybersecurity label is shared with experts in Phase 4 (communicating the results). This label is dynamic and varies based on the certification results of the system. Note that it is not mandatory to display the label on the certified product. The manufacturer is responsible for deciding whether to add the label on the certified product.

3.5.3 Certificate

SURFACE uses content and format from the EUCC for generating the certificates for the certified system. These certificates should contain the specified information in the following format,

1. an ID that is unique to the certified target,
2. information about the certified target,
    (a) name of the certified product,
    (b) type and version (if applicable),
    (c) manufacturer,

(d) link to access supplementary security information of the system based on CSA Article 55;

3. technical information of the certified target
   (a) authority name and contact information that issued the certification,
   (b) accredited third party lab information who performed the testing and risk assessment, if it is different from the authority body,
   (c) certification standards or schemes involved and its version,
   (d) assurance level: basic or substantial or high,
   (e) reference to certification report,
   (f) reference to GPP of the certified target,
   (g) reference to certification schemes or standards involved,
   (h) date of issuance and date of expiration;
4. cybersecurity label.

This format is applicable to all the ICT product certificates. SURFACE recommends generating this certificate in English when shared publicly such as on a vendor's website, or on the CC portal. If the vendor prefers, they can generate the report in their local language (along with a courtesy translation in English) and share it on their website. In addition, it is the responsibility of the issuer of the certificate to provide a guideline for the users in accessing the relevant information about the certification using the unique ID of the certified target. The certificate issuer may create a standardised format for the guidance or rules (may refer to the ENISA guidance, if required) and use the format for all the products with an automated script instead of creating individual guidelines for each product.

3.5.4 Extended Manufacturer Usage Description (MUD)

According to ISO 3100, "Risk treatment is a risk modification process. It involves selecting and implementing one or more treatment options. Once a treatment has been implemented, it becomes a control or it modifies existing controls". In general, the results of the security evaluation are used only to validate or certify the security of the system, missing a very valuable information that reports the security flaws that our system has and how they could be avoided during the system's operation phase. In this sense, we propose as a way to address risk treatment, the creation of a behavioural profile with some recommendations (polices) to take into account during the operation phase. This profile is intended to reduce the attack surface to the allowed behaviours and it could be also used to monitor suspicious behaviours during the operation phase. This activity interoperates and makes use of the results of the previous evaluation, as the behavioural profile is intended to be generated from the security results containing both security recommendations from the manufacturer and from the security evaluation to perform a secure deployment. In particular, we propose to base the behavioural profile on a standardized format such as the Manufacturer Usage Description (MUD), due to its flexibility and scalability.

Based on the extended MUD proposed in [24] (Figure 6, new fields in bold), we link our certification approach with the generation of an extended MUD file from the results of the security evaluation. This way, the MUD can represent information such as the key size or cryptographic algorithms to be used, and even the maximum number of connections that the device is capable of supporting to avoid a denial-of-service attack. It is intended to be a set of mitigation policies recommended from the certification process to enhance and guarantee the security of the product.

3.6 Phase 4: Communicating the Results

Next, the system evaluation results are shared with experts or researchers through the cybersecurity label. Based on the conditions specified in [1], information like feedback, suggestions about the PPs are shared and received with other experts. The certificates generated for the certified system are valid for a maximum of five years from the date of certificate issuance [1]. This initial period of validity can be extended if the certified system still meets its required security baselines. On the other hand, if any of the adverse conditions

in Table 1 occurred before the expiration of the certificate, the validity ends (and the system can undergo recertification).

All the information related to the certified system, such as the certification report, the certificate, and the cybersecurity label can be made available on the manufacturer's website, CC portal, or any other website in accordance with the CSA Article 55 and Article 50. Also, the certificate issuer is responsible for establishing the guidelines or rules on how to deliver and publish the certification data of the system. The issuer can also refer to the ENISA guidance.

The published information should be available only for five years after the expiration date of the certificate. The availability time may change if the system undergoes recertification and information is again made available for five years from the new certificate's expiration date. The information required by the ITSEF and the manufacturer for the conformity assessment like assessment samples or any sensitive data should be shared only upon request and the shared data must be stored securely. The conditions for sharing and publishing should be mentioned in the mutual recognition agreement (MRA) under appropriate sections.

A platform that collects and analyses different certificates and connections between them (Seccerts)[14] is described in Deliverable 7.7. The structured certification report and cybersecurity label make the evaluation result machine readable and analysable and help in discovering dependencies between certificates. Should a new vulnerability occur and a certificate be revoked, the change can automatically be propagated and other affected certificate holders notified.

[14] https://seccerts.org

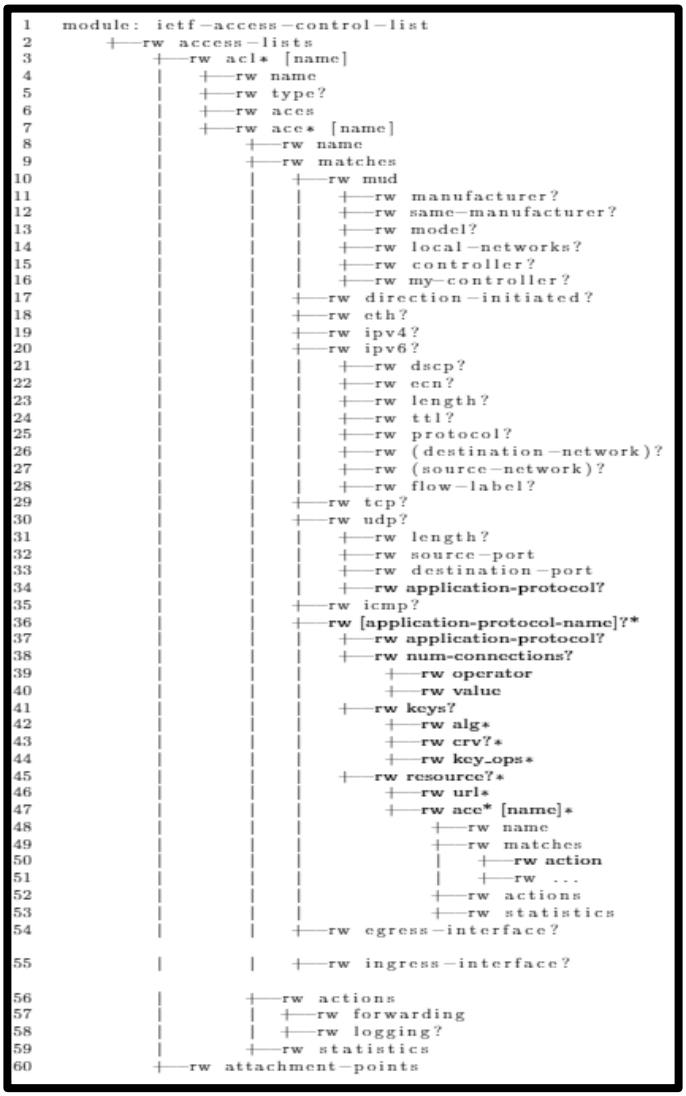


Figure 6: Extended MUD file

3.7 Phase 5: Recertification

Incremental certification is the process of identifying and evaluating the parts of the system that need to be recertified instead of re-evaluating the whole system. Lack of certification support for the life-cycle of the product and the high cost are the issues considered by our integrated approach. These two considerations are taken from an ENISA survey [22]. Generally, the recertification of any system has a high cost and is time-consuming. Hence, with the help of continuous monitoring, we can identify the changes and recertify only the necessary parts. This can reduce the impact on other business processes, effort and overall cost. This is possible only when there is a minor change or update. If any major update or change or identification of a new threat or vulnerability occurred, then recertification of the entire system is necessary.

A safety case (SC) represents proof or evidence ensuring that a system is safe to use in the given environment through a set of organised arguments. As mentioned in Section 3.3, the EG is required to define the modular SC for the system enabling isolation between modules (with agreed interfaces) and reuse. By using a modular SC, SURFACE helps in certifying only the specific component that requires recertification instead

of recertifying the whole system (unless required). If there are interdependencies between components then all the affected components are recertified. Thus, SURFACE can reduce the cost and effort required to recertify the system. We have defined the following steps for the expert group by adopting terms and processes like SC, dependencies, relationships, safety argument from [23] and integrated them with SURFACE to achieve modular and incremental certification.

Step 1. During context establishment Phase 1, the EG already analysed the system functionalities and its life-cycle. This helps in understanding the changes carried out on the system and in developing the SC.

Step 2. Initially, the EG identifies why, how and what has been changed. Then the EG compares the change scenario that happened on the system with various scenarios. In case a new vulnerability was detected, the EG should consider all the profiles that are related to the new vulnerability.

Step 3. If the change is major, then the system undergoes the whole reassessment. Otherwise, the EG proceeds with the following steps.

Step 4. The EG defines the SC architecture for the system. The SC modules are derived by the EG based on the level of cohesion and coupling, module interfaces and level of abstraction (information hiding). The EG identifies dependencies among each module and with the environment through dependency-guarantee relationships (DGR). DGRs can be represented using software elements involved in the system design. A dependency-guarantee contract (DGC) is identified by the EG if required. The DGC defines the relationship between the software elements. Correctness and completeness of the DGR and the DGC decide the validity of the safety argument. Hence, manual generation of the DGR is suggested for a higher assurance level.

Step 5. A safety argument (SA) is generated by the EG for each SC module and mostly uses appropriate DGRs to show that the dependency of one module is supported by another module. Now the EG links the SA modules to represent the system processes. These SA modules are integrated through the DGC defined in the SC. The advantage of using this SC contract is that modules are not linked directly. Here a module (which requires support) is linked to the SC contract which then identifies the appropriate module that is ready to support that dependency. By this, changes in the module do not reflect on the indirectly linked modules. The EG integrates all SC modules within the SC by mapping all the dependencies generated for each module.

Step 6. With the help of the SC, the EG has to identify all the profiles that are related to the changes that have been detected. Where applicable, new profiles can be derived based on the context.

Step 7. The EG assesses the change through impact and acceptable risk levels on that profile for all applicable vulnerabilities. For newly identified vulnerabilities, acceptable risk levels are generated by the EG for the profiles (as discussed in Section 3.3).

Step 8. All the requirements from steps 1-7 are accomplished and given as input to the planning phase to update the Generalised Protection Profile and the security target. Compliance is verified against CC requirements (discussed in Section 3.3). All the previous assessment reports and required documents can be shared with ITSEF under the EUCC for assessing the security strength of the modified system. Then ITSEF carries out the assessment and provides proper evidence. Based on the evidence, the certificate authorising scheme decides whether to provide a certificate to the system. Finally, the results are shared and a common unified report is generated. These are covered in Phases 2, 3 and 4 of SURFACE. Note that if more than one change scenario occurred, scenarios are prioritised based on the significance of the scenario or impact that can be caused by the vulnerability if exploited.

3.8 Continuous Monitoring

Continuous monitoring (CM) is required throughout and after the certification process. The fundamental goal of CM is to support risk management and recertification. Also, through CM we can enable proper maintenance and periodically verify the certificate validity of the system. We can use CM to verify whether the certification or recertification of the system complies with the guidelines of the EUCC. SURFACE includes continuous monitoring and generation of status reports. Based on the NIST SP 800-137 [4], SURFACE defines the monitoring process as follows.

Step 1. An Information Security Continuous Monitoring (ISCM) strategy with respect to context establishment is defined to allow the reuse of processes or information in future. We need to provide the required information about the target for the strategy. This information includes details about assets, previous and up-to-date vulnerabilities, threats, acceptable risk levels, functionalities, associated impact, the EG, ITSEF, applicable certification schemes or standards, ROE, GPP, the MRA. For instance, a repository can be maintained where all the information is stored or links to public vulnerability databases (like NIST Vulnerability Database) should be provided to look for the required information.

Step 2. The metrics are determined and the frequencies are set for monitoring and reporting. Metrics are structured information developed to support risk management decisions or helpful in generating status reports. The metrics can be derived from assessment result status reports, predefined vulnerabilities or other security information gathered by the EG in Phases 0 and 1 (through the manual procedure). The frequency of metric determination should be flexible. This flexibility varies based on requirements and the significance of the metric. For example, MUD can be used as a metric during monitoring as it describes the expected behaviour of the device. Finally, we develop a technical architecture that is composed of five steps. This architecture defines how the information from Step 1 is collected, how that information is stored, how the information is analysed and accessed for response and how the status reports are generated. In addition, this architecture helps in understanding the overall workflow of monitoring and its interoperability.

Step 3. The technical architecture (five steps) is implemented and monitoring is initiated. SURFACE allowed the use of any relevant tools during data collection or analysis if required. Assessments should be conducted, related information like the evaluation technical report and ITSEF details are collected and stored in the repository. At this point, for instance, we can set the frequency for the generation of a status report about the assessment. This status report can contain information about the assessment completion, assessment compliance with EUCC, conflicts (if occurred). Such a report can be helpful for the EG to ensure that the target evaluation meets the compliance requirements.

Step 4. The monitoring results (metrics), status reports and other collected information are analysed and verified by the EG manually at periodic intervals. For instance, if a component requires mitigation actions, the EG is responsible for verifying whether appropriate actions are carried out by the corresponding team at a given time through the status report. The EG reports all the findings in a document after analysing and the report is stored in the repository.

Step 5. The vendor has to make appropriate decisions based on the findings made by the EG. For instance, when any profile is not fulfilled, appropriate measures are taken by the mitigation team to mitigate the corresponding exploitable vulnerability. Also, if any issues are detected on the certified system, then existing processes are put on hold until the system is subjected to recertification. Status reports should be generated when the appropriate response decisions are taken.

Step 6. If required, based on the findings and responses, we can refine the ISCM strategy in terms of visibility of information, frequency of monitoring or reporting and metric determination. These modifications are made based on the system certification requirements and enabled monitoring features.

It is possible to require policies or procedures to facilitate the steps 1-6 [4]. The CM process of SURFACE is responsible for fulfilling certain tasks, compliance requirements and conditions. We define the following (but not limited to) responsibilities and rules based on the EUCC:

- Ensure that the scope and target information are clearly defined,
- detect if a process, rule, condition or decision is not in compliance with the rules and constraints mentioned by the target manufacturer,
- detect if issues occurred on the certified target (for re-assessment),
- ensure that appropriate decisions or actions are taken based on the findings and associated impact,
- ensure that ITSEF is provided with sufficient and valid information by the target manufacturer,
- ensure that ITSEF follows the rules of engagement generated by the EG and ITSEF provides proper and valid evidence,
- ensure that certification elements including the certification report (Section 3.5.1), the cybersecurity label (Section 3.5.2), and the certificate (Section 3.5.3) are generated and maintained as described in their respective sections,
- ensure that certificate decisions of the evaluated target are taken based on Table 1,
- keep track of the public vulnerability databases (e.g., NIST Vulnerability Database) for new vulnerabilities that are relevant to the target,
- ensure that rules for validity and availability of information mentioned in Section 3.6 are followed by the target manufacturer,
- ensure that the MRA is not violated.

These conditions are considered to be the most significant as they can have a high impact on the certification activities of the system. When any deviations are found, appropriate decisions or actions must be taken based on the MRA.

To ease the process of remediation, roles and responsibilities of the person who is required to complete the task can be predefined. The guidelines for defining the roles and specifications can be taken from NISTIR [13]. With the help of monitoring and role definitions, the assignment of remediation measures can be automated. Once the system manufacturer carries out all the updates or remediation, the process of notifying ITSEF and initiating the recertification of the system can be automated. To achieve this, we need continuous monitoring so that we can keep track of all actions and also the EG should declare the components to be assessed. Then the approved ITSEF is notified to proceed with Phase 2 (assessment).

3.9 Mitigation through Certificates and Threat MUD Files

3.9.1 Dependencies between Components through the Analysis of Certificates

Security certificates contain valuable information beyond the results of the security evaluation. They usually contain references to previous versions of the component or to components that depend on the certified one and vice versa. This information can be extracted from the certificates to build a tree for each certified component and show these dependencies. Seccerts[15] does precisely this, gathering information from FIPS and CC certificates to build the dependency graph.

With this information, we can use the graph built by Seccerts to have a global vision of the affected systems if one component is compromised. However, one of the problems of this approach is that not all the services

[15] https://seccerts.org/

and components of the Internet are indeed certified. Therefore, we need to combine this approach with additional sources of information in order to create a more complete tree of dependencies.

3.9.2 Dependencies between Components through the Analysis of MUD Files

The MUD standard file describes for a specific component or service the expected network behaviour in terms of access, using ACLs. Although once deployed, the component can add additional allowed communications, the MUD provides valuable information about the links and dependencies between the different services and components. Indeed, some already available tools[16] generate graphs that visually represent these dependencies.

Figure 7 represents the dependencies of a smart fridge with hypothetical external services such as www.fridgemakers.com (manufacturer server) or updates.fridgemakers.com (update server).

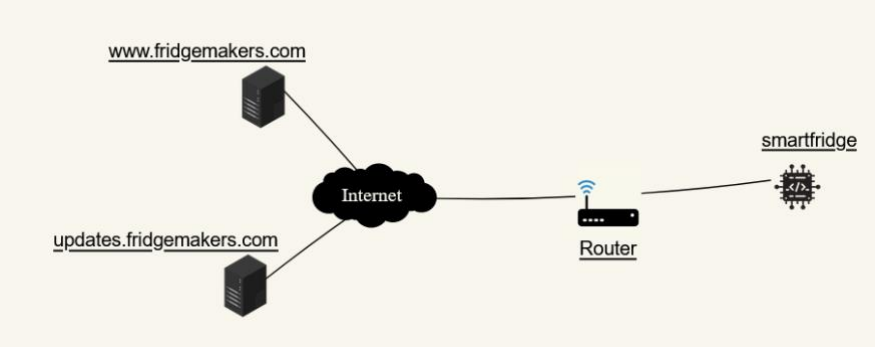


Figure 7: Example dependencies on external services (MUD visualiser tool)

By combining the information from certificates with information coming from MUD files, we can obtain a more accurate vision of the dependencies that exist between services and components (Figure 8).

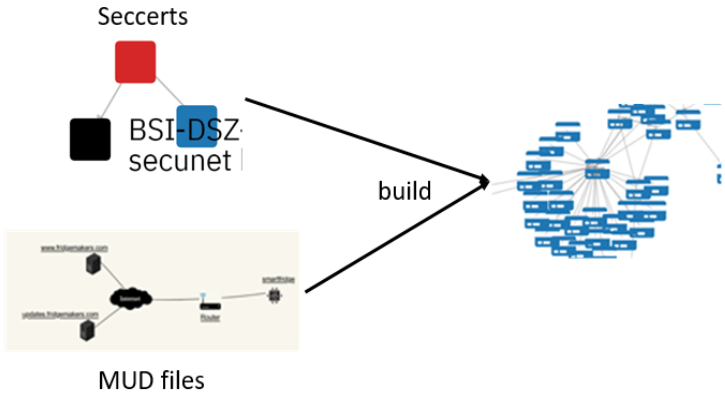


Figure 8: Dependencies between services and components

[16] https://www.mudmaker.org/mudvisualizer.php

The broader and more lightweight certification can be a good place to enforce the creation of the references between components used and in turn enable for higher utility of MUD files together with the automatic detection process. Instead of creating dependencies retrospectively by data mining the text of the certificate (as is the situation nowadays with seccerts.org tool), reliable references can be made mandatory for the certified device or service. SURFACE includes the use of structured certification reports to facilitate this process.

3.9.3 Sharing Mitigations Using Threat MUD Files

Based on the MUD standard, the NIST proposed a threat signalling mechanism using a threat MUD[17]. This MUD follows a similar structure than the MUD standard format. However, unlike the MUD standard, it is designed as a mitigation mechanism, listing those external sites to and from which traffic should be prohibited, because the sites are associated with a given threat. It is not in the threat MUD scope to list sites with which communication should be permitted, nor provide any rule regarding local network traffic. Therefore, the threat MUD is intended to be created by a threat intelligence provider rather than the manufacturer, and its model contains certain fields to identify both the threat intelligence provider and the name of the threat that the file is associated with. The reference architecture for threat signalling provided by NIST is composed by a MUD manager, in charge of obtaining the MUD file, a threat signalling server in charge of alerting of new threats and an update server, which manages the updating of the devices.

The previous generic architecture is instantiated using the concept of the threat MUD[18]. Figure 9 shows the high-level build proposed by NIST to address threat signalling using this new MUD file, extending the typical MUD architecture.

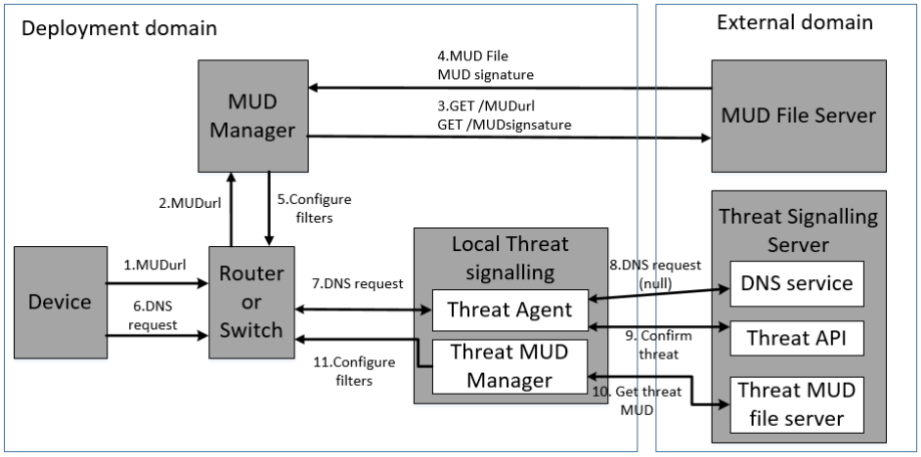


Figure 9: Threat MUD high-level build

The workflow comprises the following steps:

- The device will eventually make DNS requests to access a certain domain (step 6).

[17] https://csrc.nist.gov/publications/detail/sp/1800-15/final
[18] https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.1800-15.pdf

- The router or switch is also responsible to forward the DNS requests to the threat agent (step 7) and restrict the device communications.
- The threat agent is in charge of monitoring DNS traffic from/to devices to DNS server and detects when a DNS request is not solved (it returns a null value, step 8). When a null value is received, meaning that the domain is possibly compromised, it also asks for confirmation to the Threat API (step 9) and alerts the Threat MUD Manager about this domain.
- The DNS service, which receives information from threat intelligence providers about compromised domains, is also in charge of answering DNS queries from the device. In case the domain is marked as compromised, it returns a null value (step 8).
- Threat API receives requests from the threat agent to verify whether an unresolved domain is compromised. If so, the Threat API also gives information about the Threat Intelligence provider that marked that domain as compromised (step 9).
- The threat MUD Manager has a similar role to the usual MUD Manger. Using the compromised domain name, it will ask for the associated threat MUD file (and its signature) to the threat MUD file server (step 10). It is worth noting that the threat MUD, which is associated with a threat, will contain all the domains affected by this threat as well as the filtering rules to limit the access to them. The threat MUD Manager will also parse and enforce the filtering rules in the router (step 11).
- Finally, the threat MUD file server job will consist of storing and delivering threat MUD files associated with a compromised domain (and threat).

Although the NIST build is based on the usage of DNS traffic and currently, DNS is used only with a very limited number of certified devices within Common Criteria or FIPS 140-2 categories, MUD Manager and related tooling can be used to detect and propagate information about the potentially vulnerable referenced components even without any DNS requests – similarly as plugins into Excel spreadsheets are used to facilitate task simply because of the existing user base capable to use Excel for basic automation. For example, artificial "DNS requests" could be created for all the existing references to known vulnerable items, triggering well-known workflow users to know how to handle in the MUD Manager.

As described before, the threat MUD is intended to contain all the domains affected by a particular threat and the mitigations in terms of filtering that should be applied to the affected systems. By linking the threat MUD with the knowledge obtained before, we can share mitigations in a fast way with the services that make use of the compromised ones, analysing the dependencies tree (Figure 10).

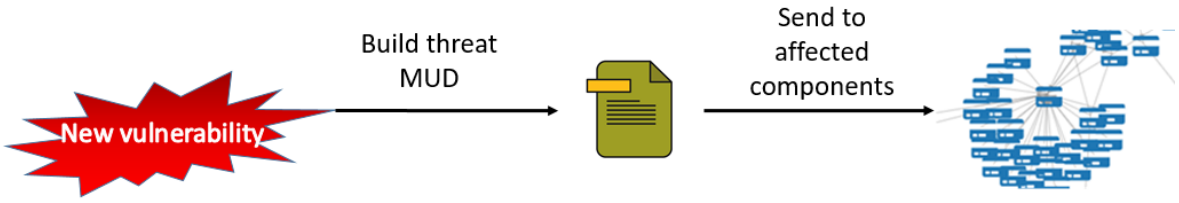


Figure 10: Sharing mitigations with services

3.10 Innovation over State of the Art

ENISA has conducted surveys to find a comprehensive set of considerations for ICT security certifications [22]. SURFACE provides a solution to three of these issues: lack of transparency, lack of certification support for the life-cycle of products, and cost and time. The lack of certification support for the life-cycle of products, and cost and time issues can be relieved by the incremental certification approach that SURFACE supports. Moreover, SURFACE covers all the steps of the lifecycle of a device, design

(certification), deployment (security configuration using MUD recommendations), operation (monitoring, mitigation and recertification), and decommissioning (revocation of the certificate).

We support the use of the cybersecurity label and a structured certification report, which introduces a layer of transparency to the certification process and results. This structured report can be used by sites that gather and analyse information about certification (e.g., Seccerts) to simplify the analysis and allow for a more reliable communication of discovered vulnerabilities and affected certificates.

4 Security Assurance Cases

Documenting and demonstrating that the primary security requirements have been designed and implemented into the system is challenging, especially when we consider the increasing size and complexity of these systems. Complex security requirements and the need to comply with security standards prescribing many requirements and intertwined processes make it difficult to gather sufficient evidence to support assurance claims and traceability for compliance checks. Structured assurance case models have become a popular tool for developing evidence-based approaches for arguing, assessing, and assuring that a specified set of critical claims regarding a system's properties are adequately justified for a given application in a given environment. To support the modular and incremental development of large and complex assurance cases, assurance case argument patterns have been proposed.

In this section, we propose a method to derive argument patterns to construct security cases from formal methods in Alloy.

4.1 Background

Assurance case models represent a reasoned and compelling argument, represented by combinations of structured claim decompositions, supporting evidence related to the system design, implementation, and development process, and a strategy demonstrating that the claim decomposition supported by the evidence will achieve the required system properties for a given application in a given environment.

4.1.1 Argument Patterns

The idea with argument patterns is that an assurance case can be developed by assembling a collection of reusable successful argument structures [25] [26]. This idea has been compared to design patterns in the area of software design. Instead, they are applied to architectures to evaluate properties such as security or safety. In this way, argument patterns are more closely related to architecture patterns.

4.1.2 Goal Structuring Notation (GSN)

There are a number of notations and existing tools for developing and documenting assurance cases, and the most popular of these is Goal Structuring Notation (GSN) [27]. GSN is a graphical notation that can be used to visualise arguments. Presenting and structuring arguments using GSN can help provide assurance of critical properties, such as those related to safety, security, and resilience of systems, services, or organisations. We adopt the GSN Pattern Notation to visualise and present the argument structure. The GSN Pattern Notation is an extension of core GSN. A summary of the graphical elements of the GSN Pattern Notation is provided in Figure 11.

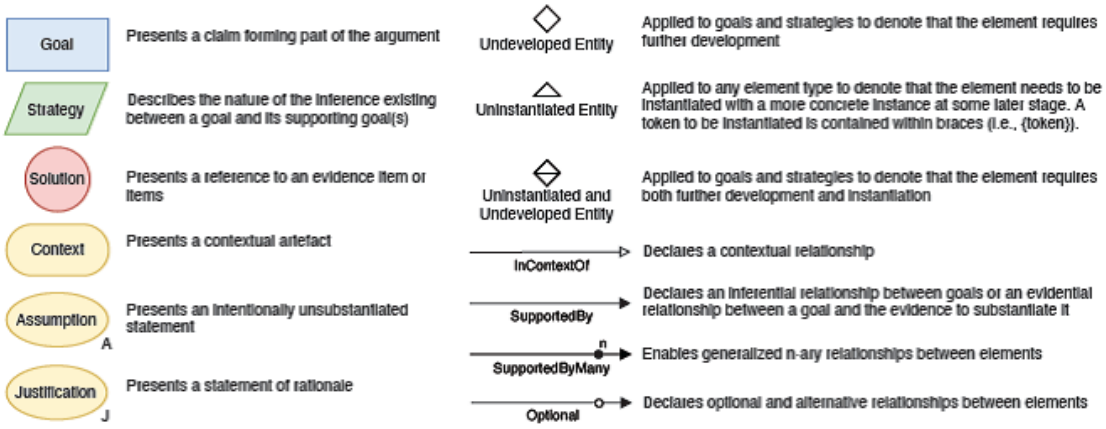


Figure 11: Principal elements of the GSN Pattern Notation

4.1.3 Alloy

Alloy [28] is a lightweight formal modelling language based on first-order relational logic. It was deeply inspired by Z mathematical notation[19] and influenced by different object-oriented modelling languages such as UML[20]. An Alloy model is composed of a set of signatures each defining a set of atoms. Atoms may have fields that define relations between atoms. In addition, signatures serve as types, and sub-typing may be defined as a signature extension. There are several ways to specify constraints in the model. One is to treat them as facts that should hold at all times. Another is to treat them as predicates defined in the form of parameterised formulas that can be used elsewhere and as assertions that are intended to follow from the facts of a model. In some situations, functions in the form of parameterised expressions may be used as helpers in the specification and verification processes.

The Alloy Analyzer [28] supports visualising models and verifying their static properties and dynamic properties (e.g., behavioural aspects). It uses a constraint solver providing automatic simulation and checking to find model instances satisfying the constraints defined during the model specification process. This makes it an appropriate candidate for our intended research work, i.e., specifying reusable security requirements libraries for software architecture design. In our work, the Alloy Analyzer essentially acts as a model checker and counter example generator. We provide the specification and verification of a representative security objective for each CIAA security objective category using Alloy. By operating as a counter example generator, when the Alloy Analyzer identifies the violation of a property, it indicates the non-fulfilment of the objective. In much the same way, by operating as a model checker, the Alloy Analyzer enables the use of a property as a policy to indicate the fulfilment of the security objective. This enables us to construct models incrementally, allowing rapid iterations between modelling and analysis when writing a specification.

The architect can instruct the Alloy Analyzer to verify whether the property prop of the system design holds, with the command:

- *check prop for n*, which would exhaustively explore every model instance within a scope of *n*, i.e., exhaustively explore every model instance to the upper bound *n* representing the number of atoms typed by each signature. If the property *prop* does not hold, a counterexample will be generated

[19] https://www.iso.org/standard/21573.html
[20] https://www.uml.org

which can be visualised. The absence of counter examples guarantees that the property holds in the modelled system, within the specified scope.

- *run prop for n*, which creates a sample of instances where the predicate *prop* holds within a scope of *n*.

4.2 Generating Assurance Cases and Argument Patterns

In this subsection, we present a set of security assurance argument patterns. These patterns provide claims to demonstrate that a system adequately satisfies its required security requirements and properties.

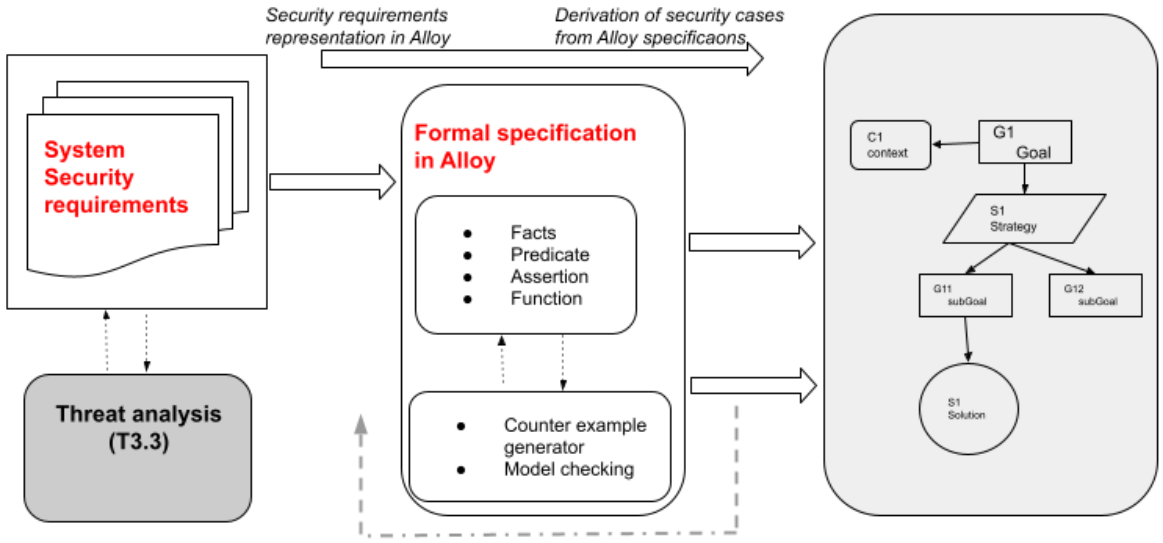


Figure 12: Derivation of security cases from formal specification

As presented in Figure 12, security assurance cases and argument patterns were derived from the proof engineering process. We based our approach on the work of [29] in the context of safety cases. The novelty of our approach is to use formal techniques to argue about security requirements (find arguing strategies) as well as finding elements of evidence. The approach is composed of two main activities: (1) representation of formalised system security requirements as formal models – through tooled formal modelling [30], and (2) derivation of security cases from the associated formal models. The dashed lines represent the accuracy of the system security requirements identification and specification. The security assurance cases belong to security objectives category classification such as CIAA and the way these requirements are formalised and verified. For instance, using the system architecture model elements such as components and connectors.

4.2.1 Incorporating Security Requirements into Formal Architecture Models

The first step in the proposed process consists of the formalisation of the system using a suitable language with automated tool support. The formal specification of the requirements depends on the specification language facilities. In other words, we can use several formal techniques and lead to different specification expressions. Consequently, different argument strategies can be derived to verify the fulfilment of the requirement in the system. The requirements that are specified using the same interpretation of the modelling elements (similar formal expressions) are supposed to be verified in the same way. As a result,

we can classify security requirements according to the way they are represented in the formal specification language. This classification will help us in the next step in building argument patterns for the security cases. For simplicity, in the following we consider only two classes:

- Class 1: Security properties that are relevant to the whole system model, referred to as global properties
- Class 2: Security properties that are relevant to some part of the system, referred to as local properties

System architecture models are described using (1) a set of structural elements, mainly components, ports and connectors, (2) required and provided services, (3) dependencies in terms of methods and connections and properties [30], [31]. Then, we (1) formally represent an interpretation of the software architecture model and the CIAA set of objectives as properties of the architecture model in Alloy. Moreover, we define how to construct from these model elements a specific predicate and assertion to be verified. Then, we will show how the verification results (e.g., counter examples and model checking results) can be used as the evidence in the associated security cases.

Example of Confidentiality objective. According to the ISO/IEC 27000:2018 standard, confidentiality denotes the property that information is not made available or disclosed to unauthorized individuals, entities, or processes. In the context of message passing communications within software architectures, confidentiality allows that the transmitted message can be obtained by other components (i.e., components can receive it) but only allowed components can get the actual content of the payload. In other words, if a component c3 other than c1 and c2 is able to get the payload d of a message m, then c1 didn't send the message m with c2 as intended receiver before that. For components c1 and c2 $\in$ C, we denote this representative property as *PayloadConfidentiality(c1 , c2)*.

The PayloadConfidentiality property is considered within the connector and can be identified by checking whether the communication system provided by the connector ensures that all information transmitted through this connector is delivered only to the intended receiver(s). The PayloadConfidentiality property is defined in this respect using Alloy as a predicate (see Listings 1 and 2).

```
sig PayloadConfidentiality extends ConnectorProperty  {}{
  payloadConfidentiality[comp1,comp2,payl]
}
pred payloadConfidentiality[c1,c2:Component, d:Payload] {
  all c3:Component-c1-c2 | let m = {m:MsgPassing | m.payload = d } |
  all t:Tick | E_get_pld[c3,m,d,t] implies
    (no t2:Tick-tick/first | no t1:t2.prevs | (H_inject[c1,m,t1] and m.
       receiver = c2)
        implies E_get_pld[c3,m,d,t2])
}
```

Listing 1. Local confidentiality property

```
assert confidentialityHold {
   (all c:ConnectorMPS | c.RestrictiveGetPld ) implies
      all c1,c2:Component, d:Payload | payloadConfidentiality[c1,c2,d]
}
```

Listing 2. Global confidentiality property

4.2.2 Derivation of Security Cases from the Associated Alloy Specification

According to the proposed classification, we will derive classification-based argument patterns. These patterns will highlight the link between the argument strategies and the logical specification of the requirements.

1. *Generic argument pattern*. We start by constructing a generic classification-based argument pattern, as shown in Figure 13. The top goal (GX.1) of the pattern is about meeting a security requirement {Requirement} of some class Class {X}. The goal is claimed in the given formal development of the system model {M} (MX.1). The proposed strategy includes arguing over the elements incorporated in the specification (SX.1) and arguing about the well-specification of the requirement in order to reduce the gap between the formal and the informal security requirements (SX.2). We rely on domain and formalisation experts' inspection (SnX.2) as the evidence that the associated model elements are appropriate formalisation of the requirement (GX.2). And we rely on the results of the model analysis (SnX.1) provided by the Alloy analyser as the evidence that the model verifies the logical property derived from the security requirement (GX.3).
2. *Pattern of the well-definedness of the formal model (architecture and properties).* Recall that the whole approach is tightly dependent on the formalisation of the system model. All the developed arguments are not valid if the system model and properties were not well defined initially. This is why we propose an argument pattern about the well-definedness of the system model, as shown in Figure 14. According to Alloy language rules, a model is inconsistent if it doesn't have any core instances. An instance is a binding of values to variables. Core instances have as their variables the signatures and their fields, and they bind values to them that make the facts and declaration constraints true. The Alloy Analyzer finds instances of a model automatically by searching within finite bounds (specified by the user as a scope).
3. *Pattern of the satisfaction of the security requirements class 1.* We propose an argument pattern for the class of security requirements that can be expressed as local properties (need to be validated at some part of the system model), as shown in Figure 15. Here *prop* stands for some local property to be verified. *Prop* is used for the interpretation of the targeted security requirement. It should be formulated as a predicate formula in the Alloy language and well-defined according to restrictions imposed on predicates in Alloy. We rely on the analysis results of the constraints resolving as the evidence to support the satisfaction of the security requirement. In this case, the Alloy Analyzer acts as a model checker.
4. *Pattern of the satisfaction of the security requirements class 2.* We propose an argument pattern for the global security requirements, i.e., properties that must be held in the whole system model (Figure 16). These properties are represented as assertion formulas in the Alloy language and well-defined according to restrictions imposed on assertions in Alloy. We rely on the analysis results

of the constraints resolving as the evidence to support the satisfaction of the security requirement. In this case, the Alloy Analyzer acts as a counter example generator.

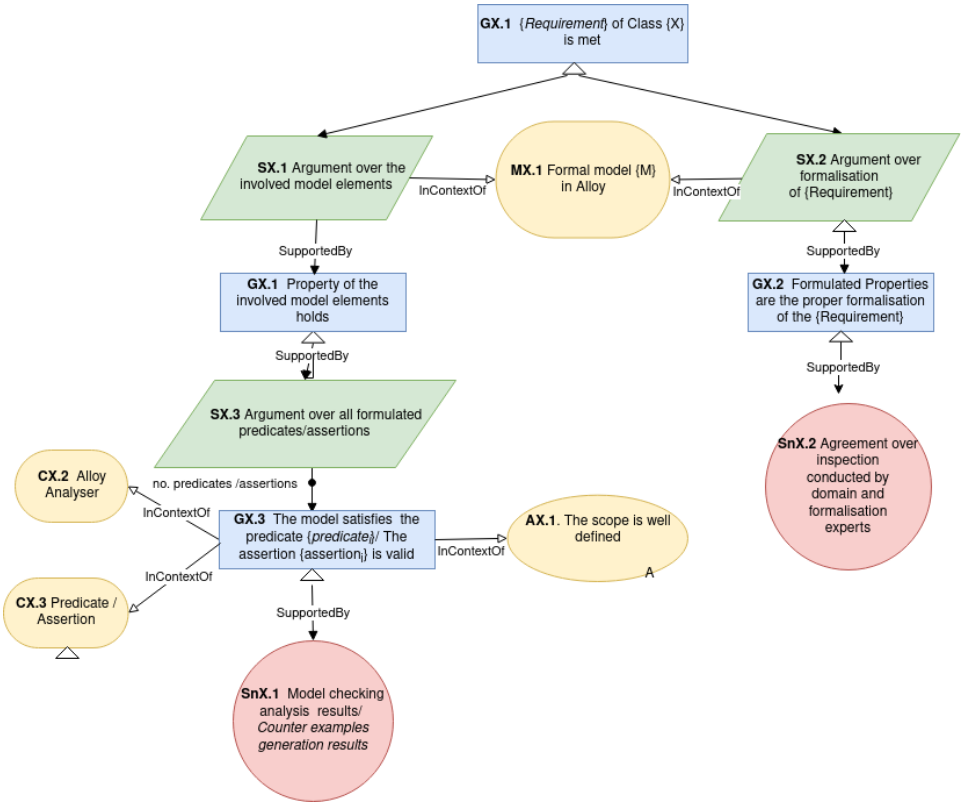


Figure 13: Generic argument pattern

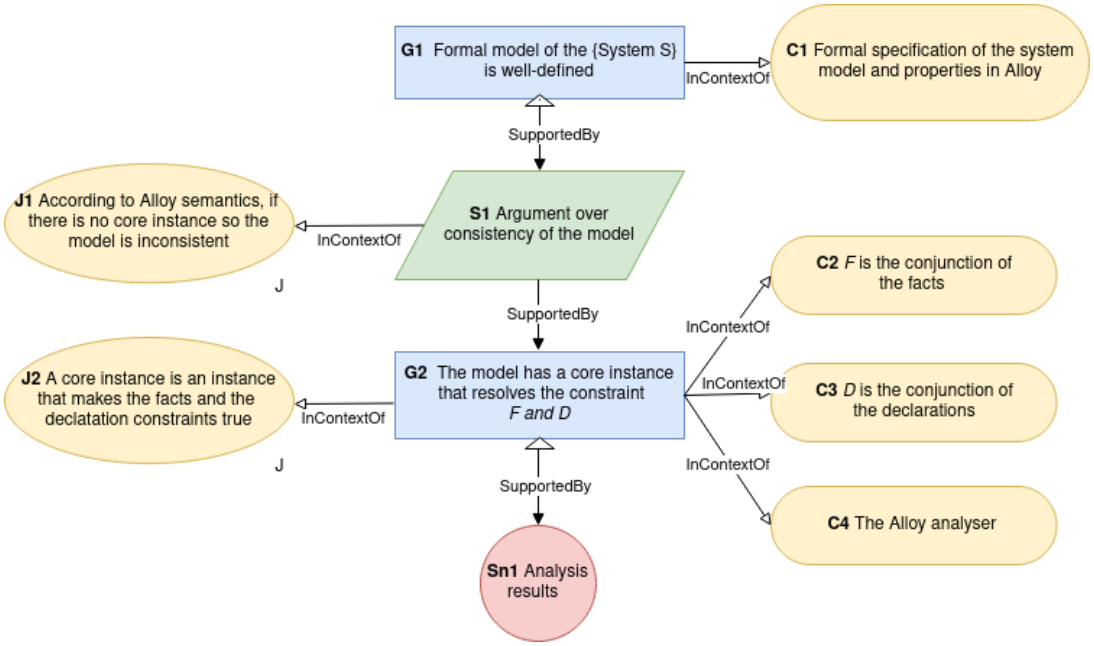


Figure 14: Argument pattern for well-definedness of formal system model and properties

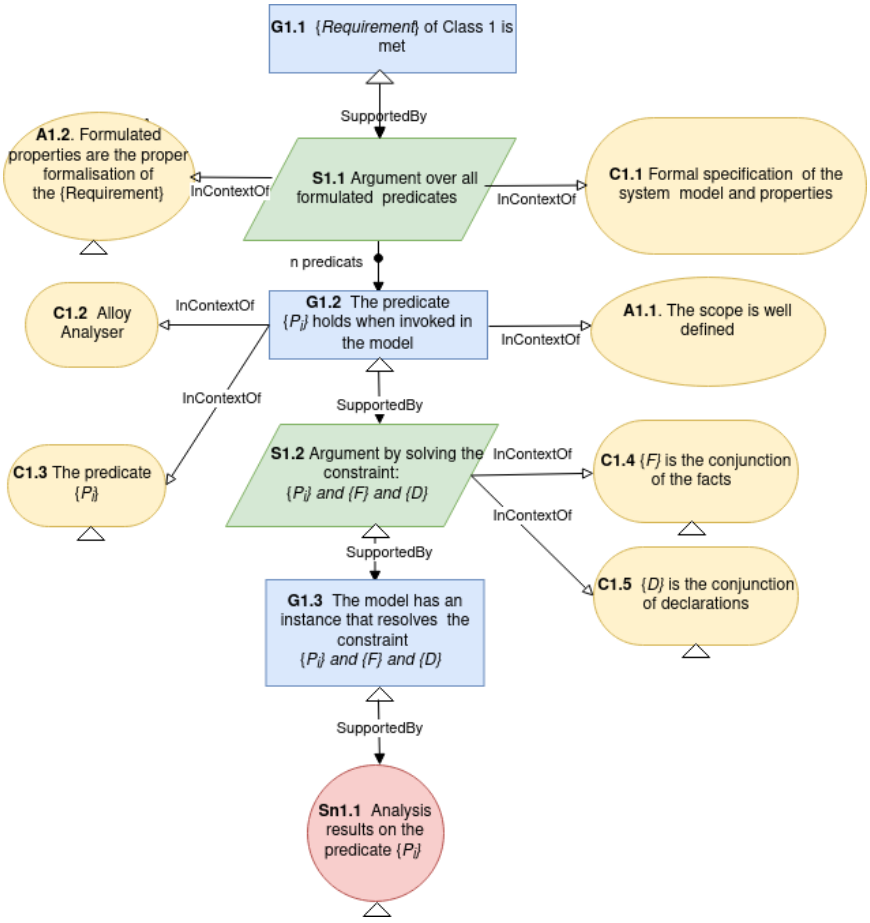


Figure 15: Argument pattern for security requirements of Class 1

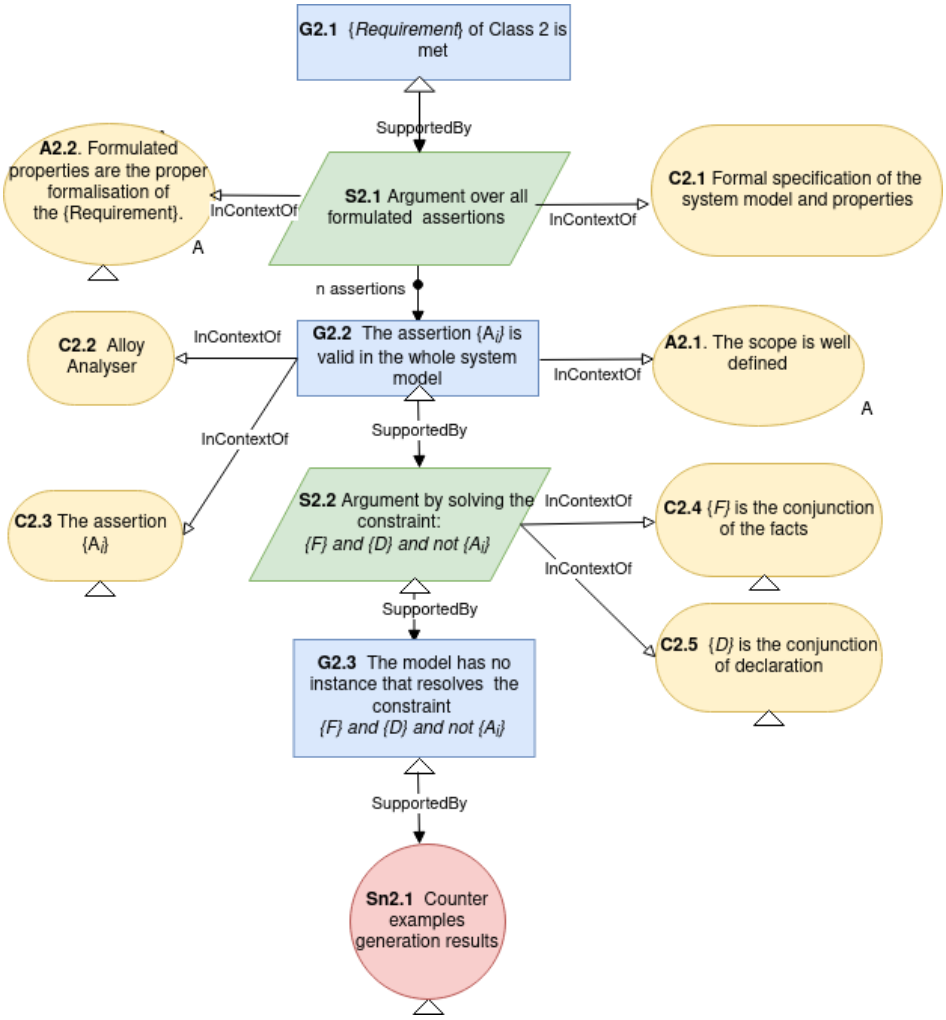


Figure 16: Argument pattern for security requirements of Class 2

5 Asset Description

5.1 Certification Assistant Tool CSA

5.1.1 Overview

CSA is a web-based asset management and certification assistant tool. We first described the CSA tool in Deliverable 3.8. Since then, it has been completely refactored and the certification workflow has been rebuilt based on SURFACE. The certification dimension adds another layer to asset management. It allows the manufacturer or vendor to carry out incremental certification based on sub-assets.

5.1.2 Demonstration Example

The tool starts with the planning phase and allows the user to describe the target of evaluation, as well as the testing methods and any known vulnerabilities that might be connected to the asset. Figure 17 shows the

information screen for the certification planning stage. At the top of the screen, there is a possibility to move to the next phase (assessment).

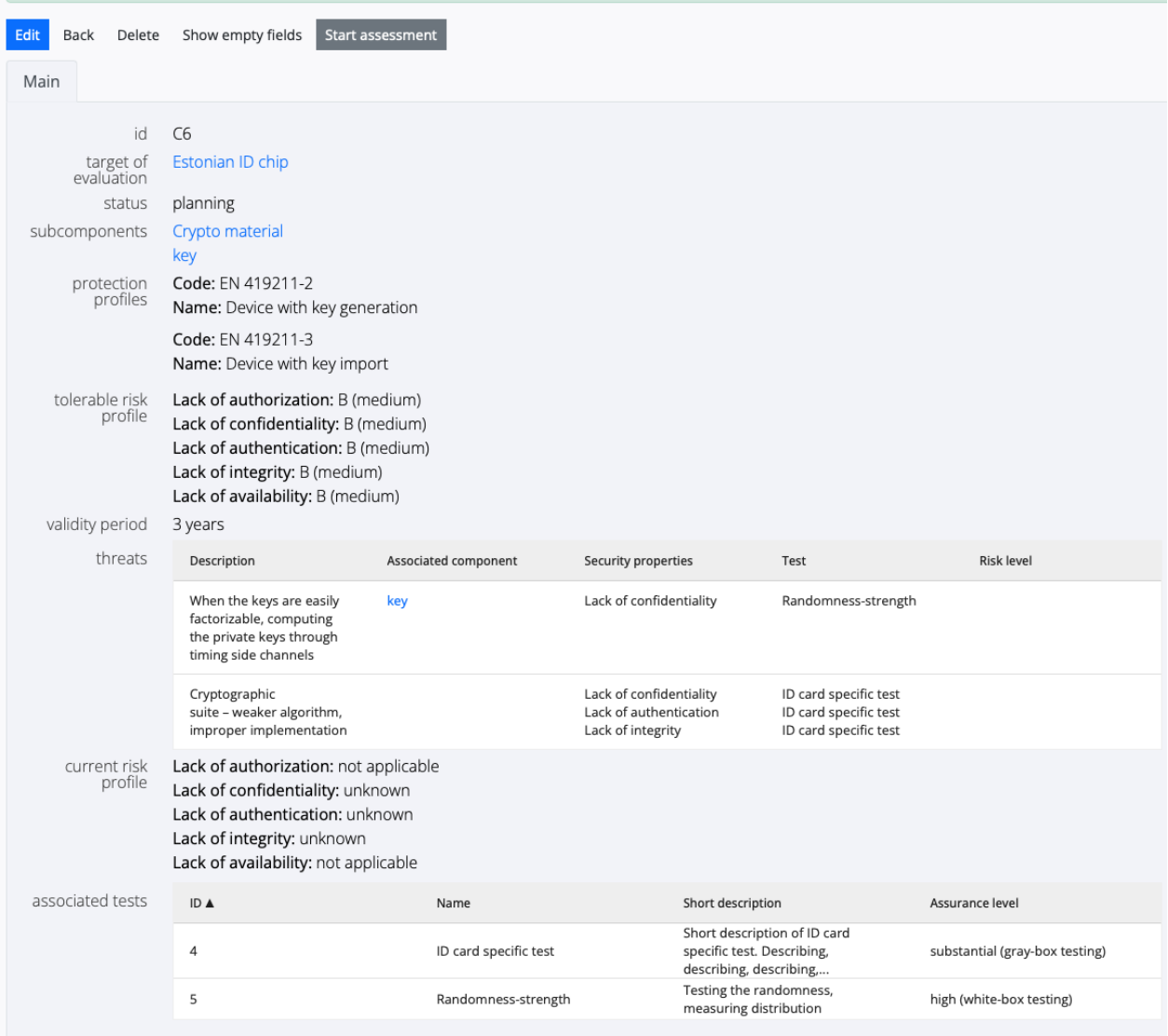


Figure 17: Certification planning in CSA

When in the assessment phase, different options for data entry become available in the edit screen. While in planning, it was possible to modify information about the TOE, then in this phase, it is only possible to enter the risk levels determined by the assessment (testing) results (Figure 18).

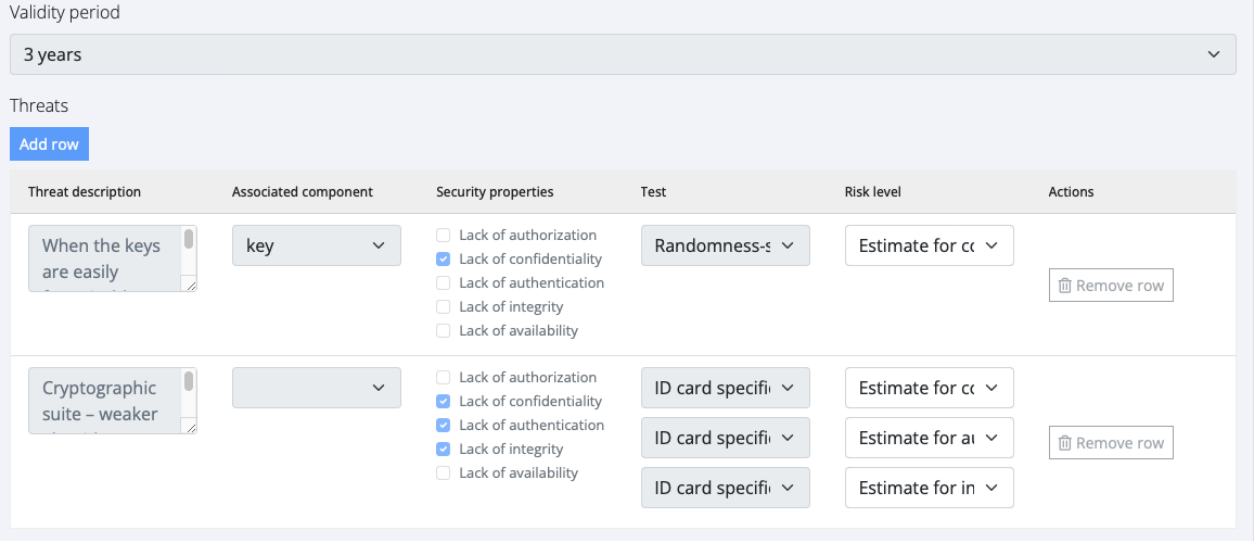
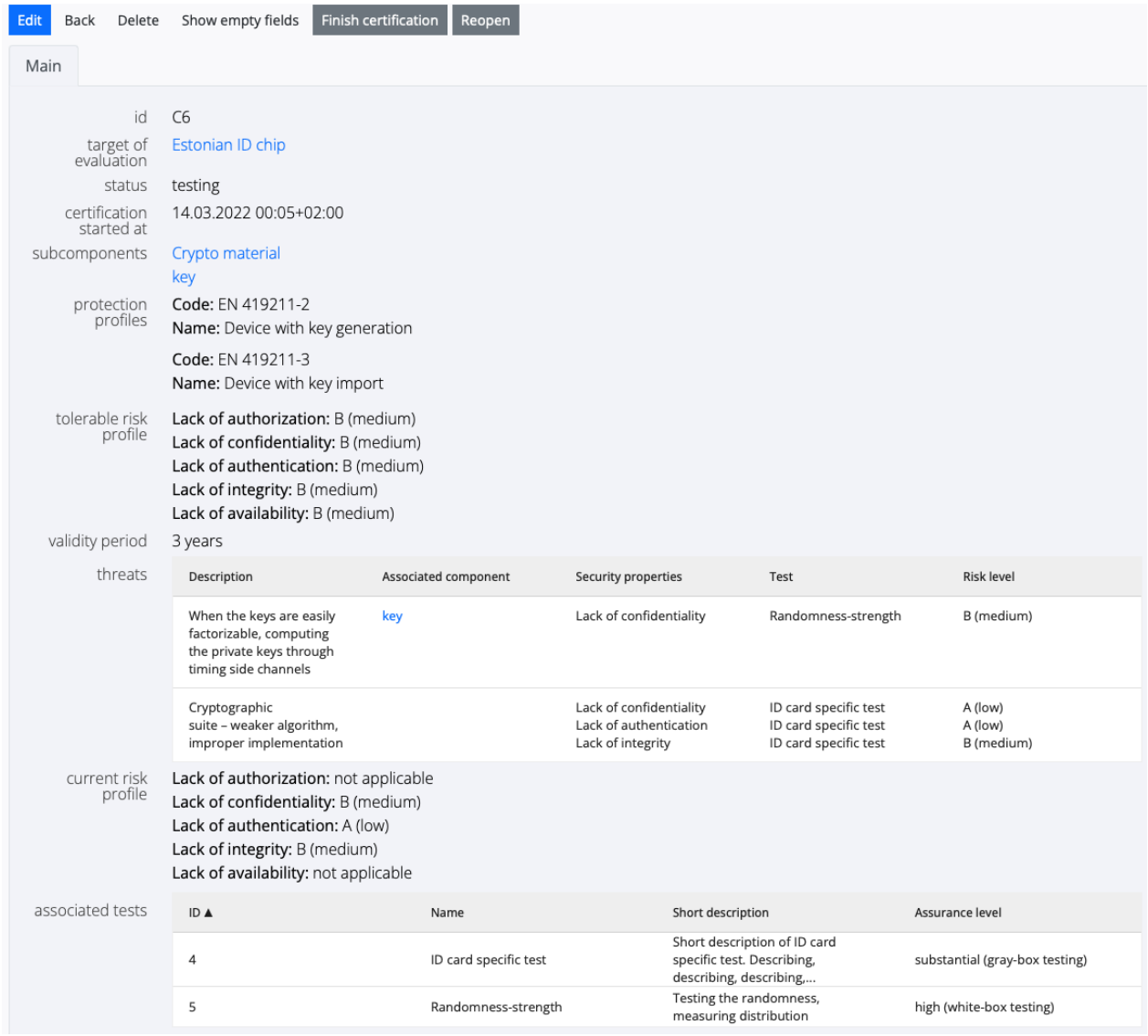


Figure 18: Assessment phase in CSA

When the assessment phase has been completed, it is possible to finish certification and the result will be determined automatically based on the user-defined tolerable risk profile and assessment results. The view before making the certification decision can be seen on Figure 19.



Edit Back Delete Show empty fields Finish certification Reopen

Main

id	C6
target of evaluation	Estonian ID chip
status	testing
certification started at	14.03.2022 00:05+02:00
subcomponents	Crypto material key
protection profiles	**Code:** EN 419211-2 **Name:** Device with key generation **Code:** EN 419211-3 **Name:** Device with key import
tolerable risk profile	**Lack of authorization:** B (medium) **Lack of confidentiality:** B (medium) **Lack of authentication:** B (medium) **Lack of integrity:** B (medium) **Lack of availability:** B (medium)
validity period	3 years

threats

Description	Associated component	Security properties	Test	Risk level
When the keys are easily factorizable, computing the private keys through timing side channels	key	Lack of confidentiality	Randomness-strength	B (medium)
Cryptographic suite – weaker algorithm, improper implementation		Lack of confidentiality Lack of authentication Lack of integrity	ID card specific test ID card specific test ID card specific test	A (low) A (low) B (medium)

current risk profile: **Lack of authorization:** not applicable; **Lack of confidentiality:** B (medium); **Lack of authentication:** A (low); **Lack of integrity:** B (medium); **Lack of availability:** not applicable

associated tests

ID ▲	Name	Short description	Assurance level
4	ID card specific test	Short description of ID card specific test. Describing, describing, describing,...	substantial (gray-box testing)
5	Randomness-strength	Testing the randomness, measuring distribution	high (white-box testing)

Figure 19: Before finishing certification in CSA

If the certification was successful, a certificate will be issued and the associated cybersecurity label together with the assurance level and QR code to the certification report will be generated (Figure 20). If the certification was not successful, the process can be started again from Phase 1. The entered information will be stored and made available for reuse.

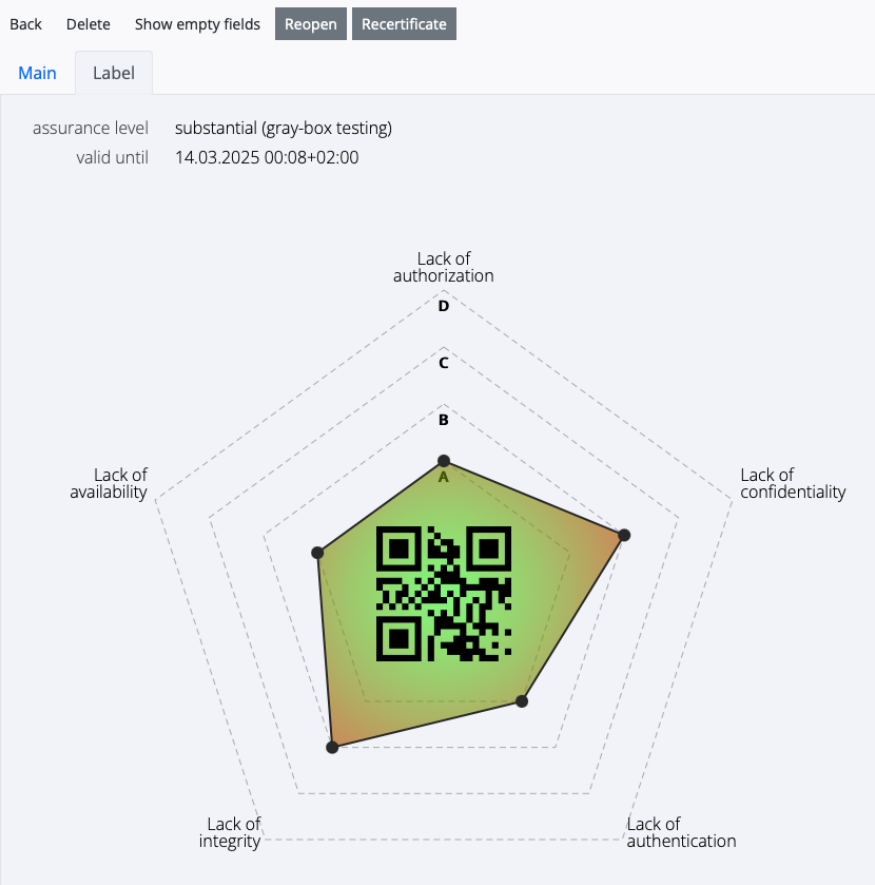


Figure 20: Generated cybersecurity label in CSA

5.1.3 Framework Components Addressed

In the administration plane of the global architecture and building blocks, the CSA tool contributes to certification and validation management. In the intelligence plane, the CSA tool falls under conformity and certification assessment. In the control and management plane, it covers security testing and certification security products.

5.1.4 Further Work

We have implemented most of the SURFACE process phases. However, we are currently missing the possibility to import different protection profiles and vulnerabilities, currently these have to be entered by the user. In addition, it would be helpful to the user if the tool included different assessment methodologies and scoring systems that could be chosen from. We also plan to add support for a detailed management of tests, for inclusion of specific test results, and in the case of model-based testing, the possibility to reference the models.

We have also been planning to add support for some specific standards, namely E-ITS[21] (the Estonian Information Security Standard) and ISO/IEC 27001[22] (Information technology — Security techniques — Information security management systems — Requirements) to support their deployment.

[21] https://eits.ria.ee
[22] https://www.iso.org/standard/54534.html

6 Conclusion

The conclusion is structured to fit the GitHub overview format.

Content

We present SURFACE – a support framework for certification which integrates and combines steps and processes from the ARMOUR methodology, the ECSO meta-scheme, the European Cybersecurity Candidate Scheme and the continuous monitoring process from NIST 800-137. The process in SURFACE is divided into phases inspired by the ARMOUR methodology: Phase 0 (reconnaissance), Phase 1 (planning), Phase 2 (assessments), Phase 3 (generating certification elements), Phase 4 (communicating the results), Phase 5 (recertification) and finally, the continuous monitoring process.

We found that each of the solutions is missing some processes. The ECSO meta-scheme approach is more general than the ARMOUR methodology and allows the certification of different products and services (not only IoT devices). However, it lacks processes for recertification or continuous monitoring. Another drawback of the meta-scheme approach is that there is no common platform or database to share the findings and results. Combining solutions from different approaches can help in creating a solution that deals with the shortcomings.

SURFACE defines the security assessments as processes that are carried out by a conformity assessment body or an accredited third-party. For risk assessment, SURFACE uses semi-quantitative risk assessment. To support risk communication, the generated risk scores (quantitative) can be mapped to the severity levels (qualitative).

SURFACE represents the certification in a structured manner. This introduces a layer of transparency to the certification process and results, simplifies automatic analysis and allows for a more reliable communication of discovered vulnerabilities and affected certificates. The framework uses a cybersecurity label similar to the ARMOUR methodology. This helps to represent the certified product as well as the security level of the security properties.

SURFACE also supports the recertification of subcomponents to reduce the time and cost of this process.

Resources:

Videos:

- https://github.com/cs4ewp3/wp3/tree/main/3.8

Scientific Dissemination:

- Jaskolka J., Jawad A., Samuel J., Hamid, B. (2021) A Security Property Decomposition Argument Pattern for Structured Assurance Case Models. In: 26th European Conference on Pattern Languages of Programs (EuroPLoP'21). Article 24, 1–10. Association for Computing Machinery, New York, NY, USA, https://doi.org/10.1145/3489449.3490001
- Quentin Rouland, Brahim Hamid, and Jason Jaskolka. Specification, detection, and treatment of stride threats for software components: Modeling, formal methods, and tool support. Journal of Systems Architecture, 117:102073, 2021, https://doi.org/10.1016/j.sysarc.2021.102073
- Quentin Rouland, Brahim Hamid, Jason Jaskolka: Formal specification and verification of reusable communication models for distributed systems architecture. Future Gener. Comput. Syst. 108: 178-197 (2020), https://doi.org/10.1016/j.future.2020.02.033

7 Bibliography

[1] ENISA, *Cybersecurity Certification: EUCC Candidate Scheme,* https : / / www . enisa . europa . eu / publications / cybersecurity - certification-eucc-candidate-scheme, 2020.

[2] ECSO, *European Cyber Security Certification, A meta-scheme approach v1.0.,* https://ecs-org.eu/documents/publications/5a3112ec2c891.pdf, 2017.

[3] "ARMOUR project. D1.1: ARMOUR Experiments and Requirements," 2016.

[4] NIST, *SP 800-137: Information Security Continuous Monitoring (ISCM) for Federal Information Systems and Organizations,* https:// nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-137.pdf, 2011.

[5] ETSI, "ETSI EG 203 251: Methods for Testing Specification; Risk-based Security Assessment and Testing Methodologies," https://www.etsi.org/deliver/etsi_eg/203200_203299/203251/01.01.01_50/eg_203251v010101m.pdf, 2015.

[6] "ISO 31000:2019 Risk management - Guidelines," https://www. iso.org/obp/ui#iso:std:iso:31000:ed-2:v1:en, 2019.

[7] "ISO/IEC/IEEE 29119-1:2013, Software and systems engineering — Software testing — Part 1: Concepts and definitions," https: //www.iso.org/obp/ui/#iso:std:iso-iec-ieee:29119:-1:ed-1:v1:en, 2013.

[8] S. N. Matheu, J. L. Hernández-Ramos, P. S. and A. F. Skarmeta, "Extending MUD Profiles Through an Automated IoT Security Testing Methodology," *IEEE Access,* vol. 7, pp. 149444-149463, 2019.

[9] A. Robles Enciso, A. Zarca, D. Garcia Carrillo, J. Hernández-Ramos, J. Bernal Bernabe, A. Skarmeta and S. N. Matheu Garcia, "Security Architecture for Defining and Enforcing Security Profiles in DLT/SDN-Based IoT Systems," *Sensors,* vol. 20, p. 1882, 2020.

[10] *NIST SP 1800-15. Securing Small-Business and Home Internet of Things (IoT) Devices: Mitigating Network-Based Attacks Using Manufacturer Usage Description (MUD),* https://www.nccoe.nist.gov/publication/1800-15/.

[11] ECSO, *State of the Art Syllabus updated,* https://ecs-org.eu/documents/publications/5a31129ea8e97.pdf, 2017.

[12] NIST, *SP 800-53A Rev. 4, Assessing Security and Privacy Controls in Federal Information Systems and Organizations: Building Effective Assessment Plans,* https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53Ar4.pdf, 2014.

[13] NIST, *NISTIR 8011, Automation Support for Security Control Assessments, Volume 1: Overview,* https://nvlpubs.nist.gov/nistpubs/ir/2017/NIST.IR.8011-1.pdf, 2017.

[14] ECSO, *European Cyber Security Certification, Assessment Options*, https://ecs-org.eu/documents/publications/5ea49d3a940a3.pdf, 2019.

[15] M. Felderer, M. Büchler, M. Johns, A. D. Brucker, R. Breu and A. Pretschner, "Chapter One - Security Testing: A Survey," in *Advances in Computers. Vol 101*, Elsevier, 2016, pp. 1-51.

[16] I. Schieferdecker, "Model-Based Testing," *IEEE Software*, vol. 29, pp. 14-18, 2012.

[17] W. Li, F. Le Gall and N. Spaseski, "A Survey on Model-Based Testing Tools for Test Case Generation," *Tools and Methods of Program Analysis*, pp. 77-89, 2018.

[18] M. Bishop, "About Penetration Testing," *IEEE Security & Privacy*, vol. 5, pp. 84-87, 2007.

[19] ISECOM, *The Open Source Testing Methodology Manual (OSSTMMv3)*, https://www.isecom.org/OSSTMM.3.pdf, 2010.

[20] PTES, *The Penetration Testing Execution Standard Documentation Release 1.1*, https://buildmedia .readthedocs.org/media/pdf/pentest-standard/latest/pentest-standard.pdf., 2017.

[21] NIST, *SP 800-30, Guide for Conducting Risk Assessments*, https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-30r1.pdf., 2012.

[22] ENISA, *Considerations on ICT security certification in EU: Survey Report*, https://www.enisa.europa.eu/publications/certification_survey, 2017.

[23] J. Fenn, R. Hawkins, P. Williams, T. Kelly, M. Banner and Y. Oakshott, "The Who, Where, How, Why And When of Modular and Incremental Certification," in *2nd IET International Conference on System Safety*, 2007.

[24] S. N. Matheu, J. L. Hernández-Ramos, S. Pérez and A. F. Skarmeta, "Extending MUD Profiles Through an Automated IoT Security Testing Methodology," *IEEE Access*, vol. 7, pp. 149444-149463, 2019.

[25] R. Hawkins and T. Kelly, *A Software Safety Argument Pattern Catalogue. Technical report*, University of York, 2013.

[26] J. J., J. A., S. J. and B. Hamid, "A Security Property Decomposition Argument Pattern for Structured Assurance Case Models," in *26th European Conference on Pattern Languages of Programs (EuroPLoP'21)*.

[27] GSN Working Group, *GSN Community Standard Version 2*, https://scsc.uk/r141B:1?t=1, 2018.

[28] *Alloy analyzer*, http://alloytools.org/, 2019.

[29] Y. Prokhorova, L. Laibinis and E. Troubitsyna, “Facilitating construction of safety cases from formal models in event-b,” *Information and Software Technology,* vol. 60, pp. 51-76, 2015.

[30] Q. Rouland, B. Hamid and J. Jaskolka, “Specification, detection, and treatment of stride threats for software components: Modeling, formal methods, and tool support,” *Journal of Systems Architecture,* vol. 117, p. 102073, 2021.

[31] Q. Rouland, B. Hamid and J. Jaskolka, “Formal specification and verification of reusable communication models for distributed systems architecture,” *Future Gener. Comput. Syst,* vol. 108, pp. 178-197, 2020.