



BSI Standards Publication

**Security Evaluation Standard for IoT
Platforms (SESIP) — An effective methodology
for applying cybersecurity assessment
and re-use for connected products**

National foreword

This British Standard is the UK implementation of EN 17927:2023.

The UK participation in its preparation was entrusted to Technical Committee IST/33/3, Security Evaluation, Testing and Specification.

A list of organizations represented on this committee can be obtained on request to its committee manager.

Contractual and legal considerations

This publication has been prepared in good faith, however no representation, warranty, assurance or undertaking (express or implied) is or will be made, and no responsibility or liability is or will be accepted by BSI in relation to the adequacy, accuracy, completeness or reasonableness of this publication. All and any such responsibility and liability is expressly disclaimed to the full extent permitted by the law.

This publication is provided as is, and is to be used at the recipient's own risk.

The recipient is advised to consider seeking professional guidance with respect to its use of this publication.

This publication is not intended to constitute a contract. Users are responsible for its correct application.

© The British Standards Institution 2023
Published by BSI Standards Limited 2023

ISBN 978 0 539 23987 4

ICS 35.030; 35.240.95

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 30 November 2023.

Amendments/corrigenda issued since publication

Date	Text affected
------	---------------

English version

**Security Evaluation Standard for IoT Platforms (SESIP).
An effective methodology for applying cybersecurity
assessment and re-use for connected products.**

Norme d'évaluation de la sécurité pour les plates-
formes IoT (SESIP) - Une méthodologie efficace pour
appliquer et réutiliser des évaluations de la
cybersécurité de produits connectés

Sicherheitsbewertungsstandard für IoT-Plattformen -
Eine effektive Methode zur Anwendung der
Cybersicherheitsbewertung und Wiederverwendung
für vernetzte Produkte

This European Standard was approved by CEN on 13 April 2023.

CEN and CENELEC members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration. Up-to-date lists and bibliographical references concerning such national standards may be obtained on application to the CEN-CENELEC Management Centre or to any CEN and CENELEC member.

This European Standard exists in three official versions (English, French, German). A version in any other language made by translation under the responsibility of a CEN and CENELEC member into its own language and notified to the CEN-CENELEC Management Centre has the same status as the official versions.

CEN and CENELEC members are the national standards bodies and national electrotechnical committees of Austria, Belgium, Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Republic of North Macedonia, Romania, Serbia, Slovakia, Slovenia, Spain, Sweden, Switzerland, Türkiye and United Kingdom.



**CEN-CENELEC Management Centre:
Rue de la Science 23, B-1040 Brussels**

Contents	Page
European foreword	3
Introduction	4
1 Scope	5
2 Normative references	5
3 Terms, definitions, symbols and abbreviated terms	5
4 Overview	6
5 Security Functional Requirements (SFRs)	19
6 Security Process Packages (SPPs)	38
7 Security Assurance Requirements (SARs)	40
8 SESIP Assurance Levels	53
Annex A (informative) SESIP evaluation case example	60
Annex B (informative) Guidance — Attack potential rating	61
Annex C (informative) Example use cases	64
Annex D (informative) Security Target template	73
Annex E (Normative) Composition Guidelines	92
Annex F (Informative) SESIP in overall product securing process	98
Bibliography	101

European foreword

This document (EN 17927:2023) has been prepared by Technical Committee CEN/JTC 13 “Cybersecurity and Data Protection”, the secretariat of which is held by DIN.

This European Standard shall be given the status of a national standard, either by publication of an identical text or by endorsement, at the latest by May 2024, and conflicting national standards shall be withdrawn at the latest by May 2024.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. CEN shall not be held responsible for identifying any or all such patent rights.

Any feedback and questions on this document should be directed to the users’ national standards body. A complete listing of these bodies can be found on the CEN website.

According to the CEN-CENELEC Internal Regulations, the national standards organisations of the following countries are bound to implement this European Standard: Austria, Belgium, Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Republic of North Macedonia, Romania, Serbia, Slovakia, Slovenia, Spain, Sweden, Switzerland, Türkiye and the United Kingdom.

Introduction

This document specifies the Security Evaluation for Secure IoT Platforms (SESIP). It includes general requirements for Security Functional Requirements (SFRs), Security Process Packages (SPPs) and Security Assurance Requirements (SARs) designed to be used in the evaluation and certification of IoT platforms.

SESIP is a methodology for the security evaluation of platforms on which connected products are based. The term “platform” in SESIP is defined as the implementation of underlying features for an application layer; a platform can be subdivided in “platform parts”.

SESIP does not address the final connected product itself, but the results of the SESIP evaluation of connected platforms are meant to be able to be used as evidence for compliance demonstration to standards addressing Connected Products.

This makes SESIP not redundant with current IoT standards but a tool on which those standards can base on by reusing outputs. It is indeed impossible for a product vendor to provide, with reasonable effort, assessment evidences for all platform parts integrated from different developers/manufacturers.

This SESIP methodology specific goals are summarized below:

- To be accessible to applicable IoT products stakeholders;
- To provide clear but harmonized security claims;
- To consider time-to-market needs by providing an optimized and efficient methodology;
- To enable the reuse of evaluation results in different products and/or between different standards and avoid redundant evaluations of same platform (parts) without added value;
- To support Connected Products compliance demonstration to Connected Product standards.

Fulfilling of these goals allows SESIP raising the overall security in IoT ecosystems by increasing the number of security evaluations through clarity in security claims and optimized efforts.

1 Scope

This document specifies a cybersecurity evaluation methodology, named SESIP, for platforms and platform parts of connected IoT products. Security claims in SESIP are made based on the security services offered by those platforms. Platform parts can be in hardware and software. SESIP aims to support comparability between and reuse of independent security evaluations. SESIP provides a common set of requirements for the security functionality of platform parts which apply to the foundational platforms of devices that are not application specific. The methodology specifies the re-use of evaluation results.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 17000:2020, *Conformity assessment — Vocabulary and general principles*

ISO/IEC 17065:2012, *Conformity assessment — Requirements for bodies certifying products, processes and services*

3 Terms, definitions, symbols and abbreviated terms

For the purposes of this document, the terms and definitions given in ISO/IEC 17000:2020, ISO/IEC 17065:2012 and the following apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

- IEC Electropedia: available at <https://www.electropedia.org/>
- ISO Online browsing platform: available at <https://www.iso.org/obp>

3.1

composite platform

platform integrating a certified platform (part)

3.2

connected application application

overall software layer implementing an IoT end-user use case based on the underlying connected platform

3.3

connected application part application part

subset of the connected application defined by a specific context (e.g. data, resources, etc.) and to be isolated from the rest of the application

3.4

connected platform platform

hardware and/or software that provides secure services to a connected application

3.5

connected platform developer platform developer

developers who build platform (parts) and supply them to product vendors or to other platform developers, and who need to certify the security of the platform (parts) that they build.

3.6

connected platform part platform part part

hardware and/or software that implements a subset of the features of a connected platform, and that can be evaluated separately e.g. the hardware, a cryptographic library, an OS.

3.7

connected product product

combination of a connected platform and a connected application that a product vendor puts on the market.

3.8

keystore

repository in which certificates, private keys, or secrets can be stored.

3.9

SESIP profile

security profile generic to a type of platform (part), template for a SESIP Security Target of a platform of type targeted by the profile

3.10

SESIP Security Target SESIP ST

ST

statement of SESIP security requirements in terms of security features (SFRs and SPPs) and evaluation activities (SARs) to be addressed during the evaluation of a platform (part)

4 Overview

4.1 General

This clause provides an overview of the essential principles underlying SESIP:

- The base concepts of the methodology
- A threat model adapted to the IoT ecosystem
- A life cycle adapted to connected products in the IoT ecosystem
- Reusability, an essential objective of SESIP, in order to handle at an acceptable cost the increasing complexity of the connected platforms that need to be evaluated in the IoT ecosystem
- Accessibility, which is required to encourage product vendors to leverage security features included in evaluated connected platforms; the results of an evaluation is expected to be accessible and exploitable by security-proficient developers without the need to be evaluation specialists.
- Security self-assessment in SESIP

4.2 SESIP concepts

SESIP is originated from the ISO 15408 series ([4], [5], [6]), specialized for the evaluation of connected platforms in the context of IoT; it provides the base concepts as follows:

- SESIP keeps the main definitions and high-level concepts introduced in ISO 15408-1 [4].
- SESIP Security Functional Requirements (SFRs) for the security features to be implemented by platforms (parts) and to be evaluated; SESIP does not use the SFR catalogue specified in ISO 15408-2 [5] but keeps the concept of a catalogue of SFRs, specialized for the IoT ecosystem, but each SFR being at a level of final service to the user.
- SESIP Secure Process Packages (SPPs) for the security processes to be implemented by the developer of the platform under evaluation.
- SESIP Security Assurance Requirements (SARs) for the evaluation activities to be performed; SESIP keeps the categorization of the Security Assurance Requirements and the associated type of developer's inputs as in ISO 15408-3 [6], however it specifies again the content as described in 7.1.
- SESIP assurance levels; SESIP does not use "EAL" packages specified in ISO 15408-3 [6], but defines its own assurance packages adapted to the IoT ecosystem: the SESIP levels (see Clause 8).

See details about SESIP implementation of those concepts in Clauses 5 to 8.

SESIP is an evaluation methodology that specifies as precisely as possible how to evaluate the security of a product, in this case a connected platform. Similarly, SESIP does not specify any particular procedure, nor does it explicitly organize the mutual recognition principles between certificates, and only provides guidance and directions. A SESIP evaluation/certification scheme based on this SESIP evaluation methodology is expected to be specified in another document by the certification scheme owner.

4.3 IoT use cases and threat model

4.3.1 General

IoT is a broad term, but always contains a product ("thing") and some form of connectivity ("internet"). SESIP focuses on the "thing" side of IoT, and on the security of connected platforms, on which connected products are based.

4.3.2 Architecture

A connected platform typically includes the following components:

- Hardware (processing unit, memory, possibly a secure element, at least one network interface, possibly some sensors) and associated features e.g. Firmware, Boot Loader and Root-of-Trust.
 - It is assumed that the connected platform includes at least one network interface that is directly or indirectly connected to a network and exposed to potential attackers.
- An operating system, providing a foundation to run Connected Applications on the hardware.
- A network connectivity layer (e.g. Comm library), allowing the connection of the product to backend or other products.
- Software application services offered to connected applications, providing an application framework to product vendors (e.g. Crypto library, Secure Storage, Identity and Attestation features).

The Figure 4-1 shows an example of a Connected platform:

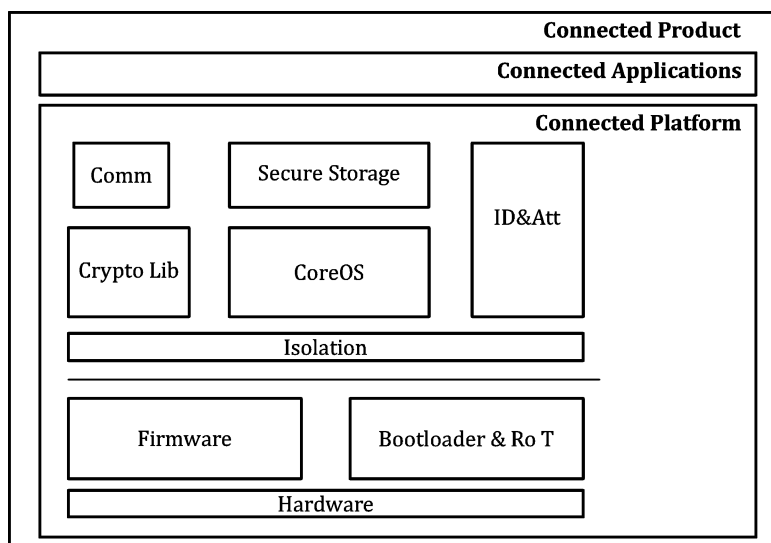


Figure 4-1 — Example of connected product architecture

4.3.3 Assets

Platform assets depend on the platform implementation and use case. However, the list of typical main assets shown in Table 4-1 has been established by a group of IoT stakeholders:

Table 4-1 — Main assets of connected platform

Asset	Protections
User data (local)	Privacy concerns are essential. Protections of confidentiality, integrity and authenticity shall be provided.
User data (authentication data)	Confidentiality is required for secrets. Secondary data (like counters) shall be appropriately protected (confidentiality, integrity).
Data in transit (internet)	Confidentiality and integrity are often essential, as is the authenticity.
Data in transit (local)	Integrity is often essential. Confidentiality is not a systematic requirement. Authenticity is less common.
Code, including platform code and application code	Integrity and authenticity are strong requirements. Confidentiality is optional.
Product identity	Integrity and unicity are required.
Configuration and system data	Integrity and authenticity are required.
Life cycle related data	Integrity is required.

It is understood that not all platforms provide complete coverage, but such limitations shall be carefully motivated when claiming SESIP SFRs (e.g. limited bandwidth or legacy protocol).

The assets may be further categorized into different criticality levels that will be protected at the appropriate level in the platform (part) – in the case of a multi-assurance platform, see 4.5.2. For instance, there may be different levels of cryptographic keys, depending on their function and life cycle. In that case:

- Protection mechanisms shall be appropriate at every level.
- Assets shall be usable without disclosing them to a lower level.
- Usage of the assets from a lower level shall be appropriately controlled (access control).

4.3.4 Attackers and threats

4.3.4.1 Base scenario

The minimum and mandatory threat model in SESIP is an attacker with only remote (no physical) access to the connected platform during the exploitation phase (see Annex B). This addresses the main IoT concern of a scalable attack exploited using a remote connection to the connected platform.

Nevertheless, the attacker can perform any type of preliminary attack on a connected platform (part) owned, including physical attacks; this shall be considered for the base scenario in an identification phase (see Annex B).

Also, in this base scenario, threats related to untrusted software that could be loaded onto connected platforms are not considered.

4.3.4.2 Extended scenario – physical access

When connected platforms are physically accessible to attackers, the threat model can be expanded and covered by the use of the SFRs “Limited physical attacker resistance” and “Physical attacker resistance”. The typical example scenarios where attackers have physical access to a victim product are:

- Connected platform deployed outside of a physically protected environment; e.g. a doorbell, outside IP camera.
- Temporary physical access; e.g. “evil maid” attacks where the attacker has temporary physical access to the product that has already been acquired by an end user, or “supply chain” attacks where the attacker delivers a compromised product to the target.

4.3.4.3 Extended scenario – untrusted software

When untrusted software can be loaded onto connected platforms, either by the end user or by an external entity, and that could impact the platform, its parts, or its applications, the base threat model can be expanded and covered by the use of the SFRs “Software attacker resistance: Isolation of platform”, “Software attacker resistance: Isolation of platform parts”, and “Software attacker resistance: Isolation of application parts”.

4.4 Connected product life cycle

Different life cycle models can be applied to connected products, and to the connected platforms that compose each product. Nevertheless, some patterns can be found in most products that are significant for security:

Vendor provisioning is the phase during which the product is provisioned with credentials that are shared with the vendor’s backend, and that allow the product to communicate securely with the backend and to perform management operations. This phase typically concludes with the delivery of the product to the customer.

User provisioning is the phase during which the product is provisioned with a user's credentials and specific data that allow the product to represent that user. This phase typically concludes with the normal usage phase of the product.

Normal usage is supposed to be the product's normal state, until one of the following events occurs:

- The user applies a factory reset, which removes all user-related data and credentials, and prepares the product to be transferred to another entity (e.g. for resale, for return, or even for temporary storage). The product is then ready again for user provisioning, but a user should not have the ability to return the product to an earlier life cycle phase.
- The user decommissions the product, before discarding it. This Terminated state is irreversible.

Some products may include an additional state related to Field return, during which specific debugging features may be available. All user data and credentials shall have been removed before reaching that state.

The product life cycle shown in Figure 4-2 is used as a reference in the SFRs when references to a life cycle are required.

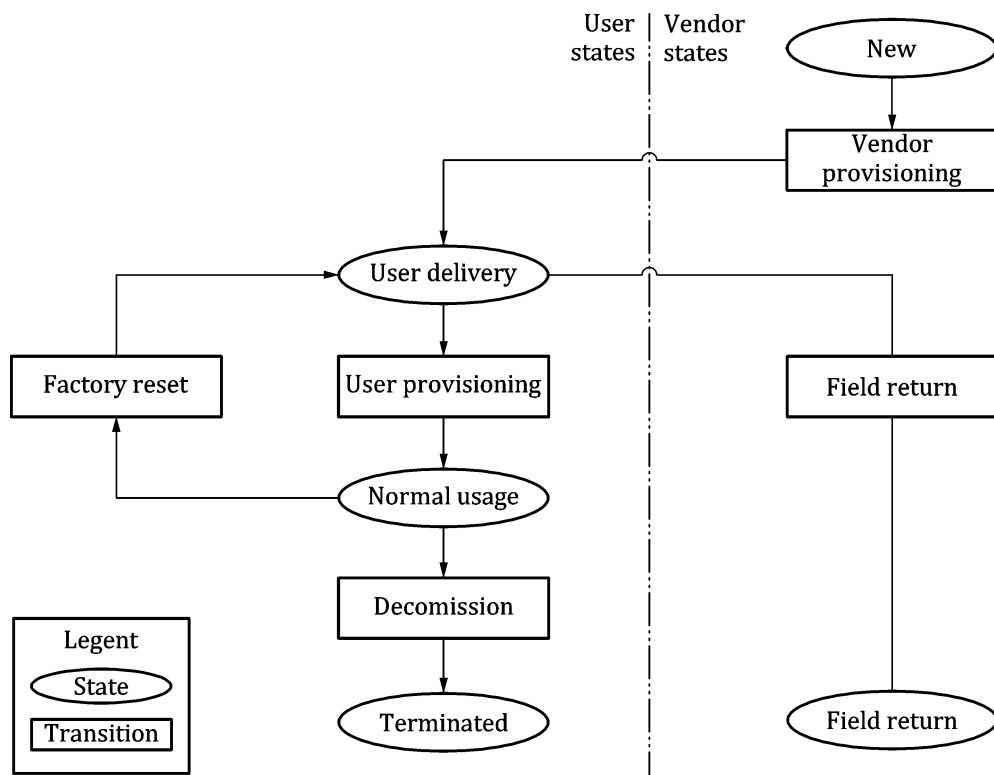


Figure 4-2 — Reference product life cycle

Note that the vendor states are only reachable by the vendor, either before delivery of the product, or after return of the product by the user.

In addition to the product life cycle, the connected platform and some of its parts may have different life cycles that are also significant to security. Such security life cycles are product-specific, and their contribution to the security of the connected platform (part) should be described in the corresponding Security Target.

4.5 Reusability in SESIP

4.5.1 General

Connected products are complex, often much more than most of the products that have had their security formally certified until today. SESIP recognizes this by providing a dedicated methodology for the connected platforms on which these products are based. Connected platforms are often built by assembling several pre-existing hardware and software components; some of them include security components that protect critical assets and need to be evaluated at a high assurance level. Such components are often integrated in several connected platforms targeting different use cases.

SESIP methodology specifies ways to independently evaluate subsets of components, which may then be called platform parts, and reuse the evaluation results in any connected platform. Those results can come from an evaluation under SESIP methodology, but also from other compatible external evaluations.

4.5.2 Building connected products from connected platforms

4.5.2.1 Reuse of external evaluations

As mentioned in 4.3, SESIP focuses on the “Things” in IoT, and more specifically on the solutions on which these connected things are built, which we call connected products. Every connected product belongs to a category or a vertical, and dedicated security standards are likely to be built for the most common types of connected products (e.g. Consumer IoT, Industrial IoT, Connected Vehicles, etc.). For each type, a specific risk analysis is needed to determine the appropriate functional and assurance requirements and may then result in the creation of a specific evaluation scheme.

In such a multi-scheme context, the SESIP methodology security requirements are specified in a way that enables the establishment of equivalence with the requirements of other schemes. After successful compatibility analysis, this allows the reuse of evaluation results between schemes.

4.5.2.2 Reuse of platform parts evaluations

4.5.2.2.1 General

A typical connected platform is not a single component, as it comprises some hardware, including at least one [micro]controller or [micro]processor, and some software, including at least an operating system. A connected platform may include many more components, for instance related to communication or security. A vendor typically builds its connected platform by selecting hardware and software components, most likely from different third-party vendors, and then assembling them.

Every vendor in that supply chain needs to provide security evaluation assessment, ending with the integrator in charge of ensuring that the fully connected platform is secure. To maintain security through the whole assembly process can be quite complex, unless all the vendors of those components (hardware and software) use a common methodology.

In order to address this, SESIP considers the following reuse contexts:

- Reuse of evaluated platform parts in several platforms (parts)
- Reuse of evaluated platform parts from a hosting platform (part) to another one

4.5.2.2.2 Reuse of evaluated platform parts in several platforms (parts)

SESIP allows the evaluation of platform parts, individually or in composition (see 4.5.3), in such a way that the evaluation results of those platform parts remain applicable in different connected products.

An example is provided in Figure 4-3:

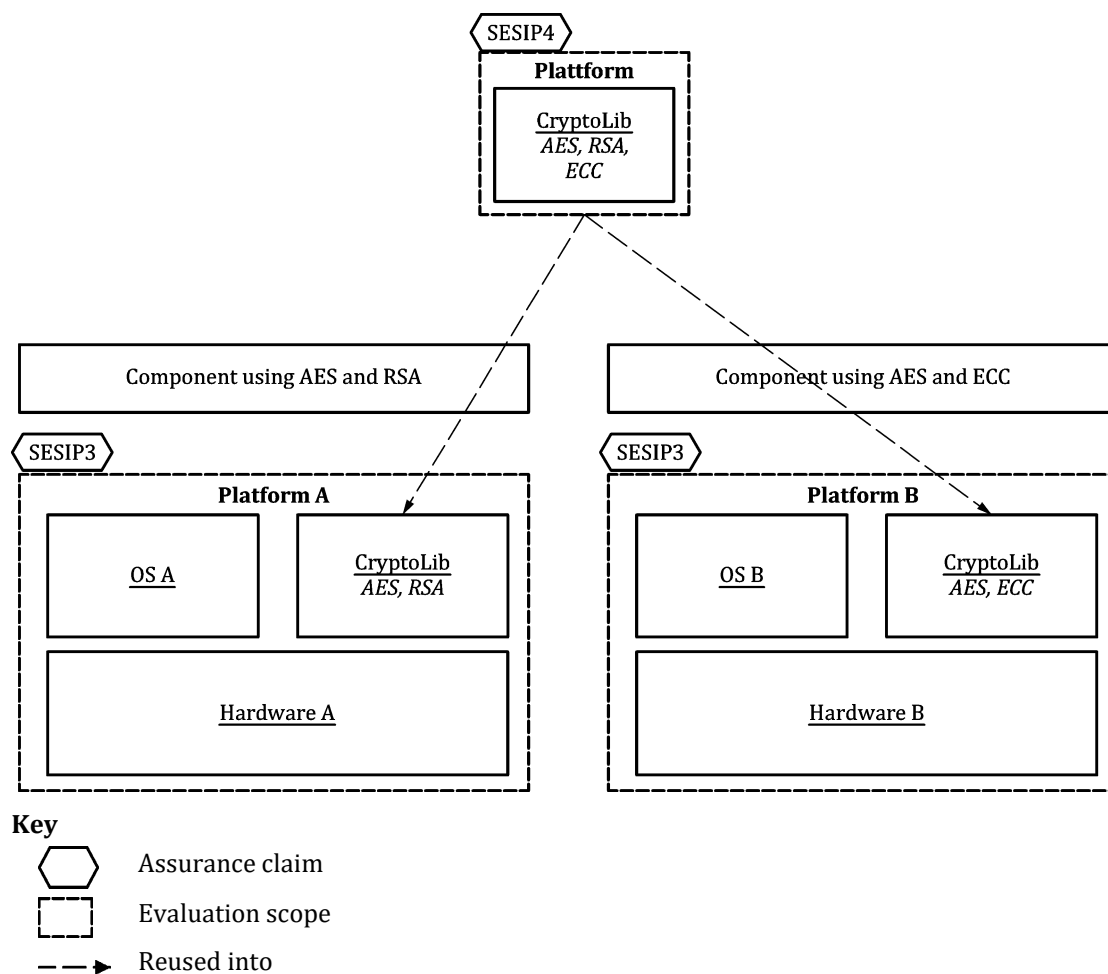


Figure 4-3 — Example of evaluation results reuse in several platforms (parts)

4.5.2.2.3 Reuse of evaluated platform parts from a hosting platform (part) to another one

SESIP allows for the reuse of evaluation results for parts evaluated inside a particular platform to be integrated into another platform (part).

An example is provided in Figure 4-4:

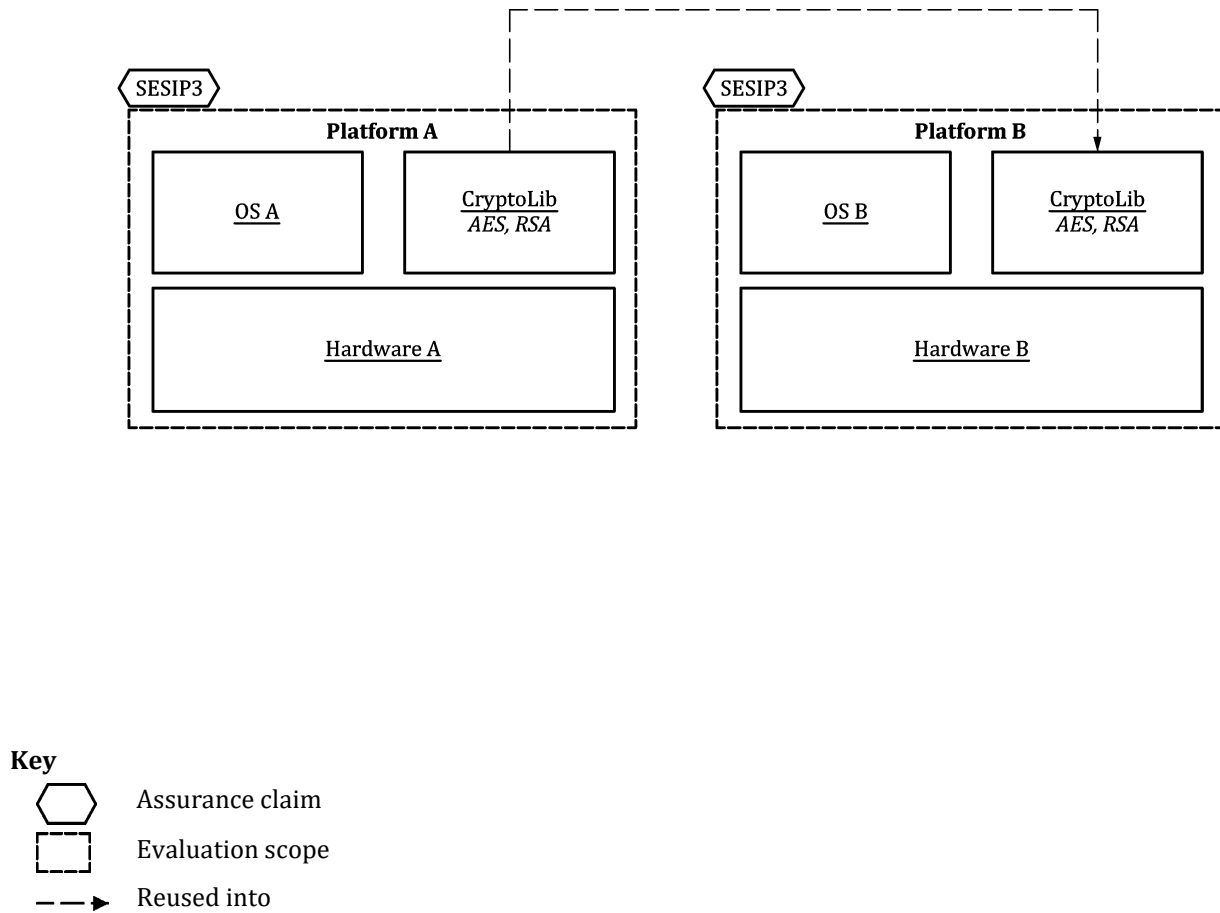


Figure 4-4 — Example of evaluation results reuse from a platform (part) to another one

In all cases, the reuse of evaluation evidence is in particular based on security integration guidance for each component; then, the assessment of the connected platforms integrating evaluated platform parts requires the verification that secure integration guidelines and composition rules (see 4.5.2) are respected.

Note that reuse effort will vary depending on the use case, from straightforward portability with few verifications, e.g. full self-containment of security functionality, to a dedicated set of activities needing to be performed on the integrating scope and to be defined in the previously mentioned integration and composition guidelines. In any case, the effort will be significantly lower than a re-evaluation on evaluated parts in a new platform (part).

Using SESIP as core methodology allows vendors to have a clear understanding of the assumptions and guidance of the third-party components they integrate by reusing evidence provided by the components' vendors during the evaluation of their platform (part).

4.5.2.3 Reuse of platform evaluation in connected products

Most Connected Products will be built by vendors who develop one or several dedicated Application(s) on top of a generic or off-the-shelf connected platform. SESIP focuses on the connected platform to provide a foundation for the security assessment, and potentially the certification, of Connected Products. It addresses the security level of the whole operating system including services used for the storage, installation, initialization, and execution of this Application, as well as its data protection.

4.5.3 Additive composition within SESIP

4.5.3.1 General

The objective of additive composition is to facilitate the evaluation of a connected platform by reusing previous evaluation results of the platform (parts).

SESIP allows different composition scenarios:

- Combination of previously evaluated platform parts
- Combination of platform part(s) to be evaluated with previously evaluated platform part(s)
- Combination of a platform to be evaluated with previously evaluated platform parts

Additive composition allows successive compositions of platform parts into wider ones which themselves can be evaluated and integrated into wider platform parts, as illustrated in Figure 4-5. Each composition benefits from the reuse of the evaluation results of the previously integrated platform part. Such integrations and related composition evaluation can be done until reaching a full IoT platform, ready to be part of an IoT device. The final IoT device can rely on the successive evaluated platform parts and reuse each evaluation result. All the supply chain actors benefit from the additive composition: from the evaluation of a hardware block, through the evaluation of intermediate modules, to an evaluated device.

An example of composition scenario is shown in Figure 4-5:

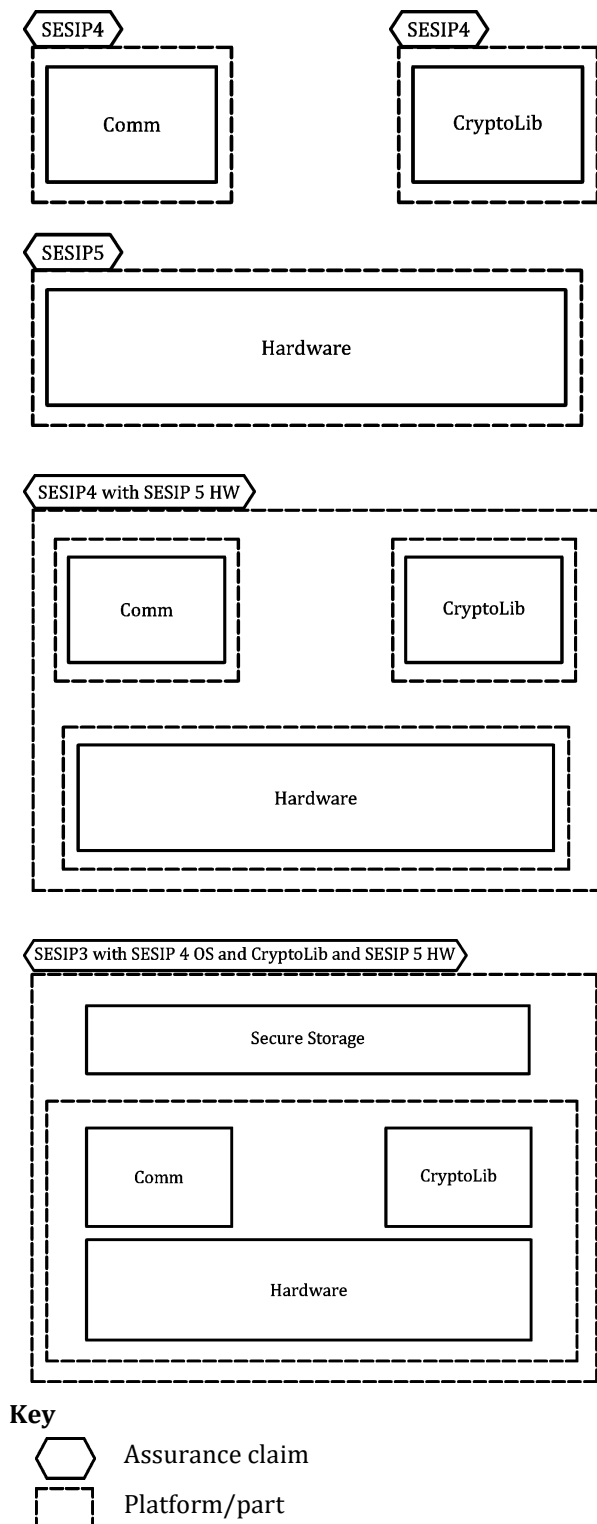


Figure 4-5 — Example of composition scenarios

4.5.3.2 Composition evaluation rules

Evaluation of a platform part to be further integrated into a wider platform (part) shall produce composition guidelines listing rules to be respected by any other platform part interacting with it. This

Individual evaluations

Any set of components aiming at being integrated into a Connected Platform can be individually evaluated

Composition 1

Each target previously individually evaluated can become a part of a composition

Composition 2

Composition 1 can then be combined with an upper layer not yet evaluated. It becomes a part of this new composition

shall be specified as an objective for the environment and referred to specifically in the guidance as per SESIP ASE and AGD requirements (see details in 7.2).

Evaluation of a composition of platform parts shall assess that the guidance (including objectives for the environment, user guidance, integration guidance, etc.) of each platform part is respected.

Evaluation of a composition of platform parts shall assess the impact of the composition on the correct security functioning of the platform parts.

Annex E outlines the steps of a composition process and provides detailed guidelines for each actor.

4.5.3.3 Assurance level of compositions

Composition may occur between platform parts that are evaluated at different assurance levels. By default, the composition can claim at most the lowest assurance level of the platform parts it is composed of. For instance, a connected platform composed of a SESIP2 platform part and a SESIP5 platform part cannot be certified at an overall level higher than SESIP2 without providing additional evidence about the SESIP2 part or about its usage by the SESIP5 platform part.

Nevertheless, in such a case, the Security Target may identify a subset of the SFRs that require higher assurance. In that case, the Security Target can claim the assurance level of the high-assurance part, provided that the SFRs covered by the higher part are clearly identified to the reader. Using the example from the previous paragraph, the connected platform could claim being at SESIP2 with a SESIP5 part covering only the SFRs of the SESIP5 part. The assurance level for such an evaluation shall be referred to as “SESIP2 with SESIP5 part(s)”.

Moreover, the SESIP level of an individual platform part could also be increased once composed with another platform part providing security features with a higher assurance level; this shall be demonstrated by evidences at the composition level.

Such multi-assurance evaluations are expected to fulfil industry requirements for composition. For instance, industry requirements may mandate that critical assets such as some cryptographic keys or credentials be specifically protected using a tamper-resistant product like a secure element, while other assets that are less critical will require a lower level of protection. In that case, the connected platform is required to have a subset evaluated to a higher assurance level, as specified above.

4.6 Accessibility and transparency

IoT security remains a challenge. A SESIP evaluation will help IoT vendors to select connected platforms that provide an appropriate level of security, but SESIP also aims to ensure that these vendors have the ability to properly leverage in their products the security features provided by evaluated platforms (parts). This is achieved mostly through the Security Target, which is expected to become a document exploitable by developers who are proficient in security but are not evaluation specialists. This can be achieved by ensuring that Security Targets are transparent and readable, that the security choices are explicit and motivated, and that SESIP can ensure some consistency between Security Targets.

To that end, SESIP Security Targets shall respect several key characteristics:

- The security objectives are the concretization of the risk analysis informing the reader about the security features to be covered by the product. They shall be specified in the Security Target writer such that developers can leverage these features, and to facilitate the reuse of SESIP evidence.
- The SESIP Security Functional Requirements (SFRs), which are the expression of the security objectives into implementation requirements, are tailored for connected platforms. In SESIP, each SFR covers a full security purpose by itself. This allows the Security Target users to develop an intuitive understanding of the security requirements. Note that flexibility for requirement construction is lower than when purposes are met by several SFRs; however, in a specific context where needs are predictable, such as a connected platform, such flexibility has been found not beneficial. SESIP specifies a set of such SFRs, which are detailed in Clause 5.

- The specification summary written by the platform evaluator shall be at a sufficient level of detail to demonstrate that the SFRs are met by the specific implementation.
- The summary description of the flaw remediation process, written by the platform (part) developer, shall explain in particular how the discovery of a vulnerability and the subsequent analysis may lead to the development and distribution of a platform update or to other corrective measures.
- Reference to user guidance shall be included, along with a description of how a prospective user can access this documentation.

Note that accessibility does not necessarily imply oversimplification; in particular, the semantics of SESIP SFRs are precisely specified, so there is no ambiguity about the meaning of the SFR, even expressed in plain language.

The SESIP Security Target and the documentation that it refers to are intended to be accessible documentation, at least to prospective adopters of the platform (part); this can be refined in individual schemes, depending on the security sensitivity of the evaluated platforms (parts).

4.7 Security self-assessment in SESIP

Security self-assessment is achievable with SESIP, but as self-assessment has many different meanings in the industry, it is important to specify it here.

Security self-assessment by the developer is based on the demonstration that all security requirements are properly covered in the form of a rationale included in the SESIP Security Target.

Such a rationale shall be provided for:

- Every Security Functional Requirement claimed, describing how this requirement is covered by the implementation
- In the case of a composition, every security objective for the environment defined by a platform part, to demonstrate how this objective is met and/or specify it again as a security objective for the platform environment
- The flaw remediation claim, providing a description of a compliant process
- Each document required by the claimed Security Assurance Level, explaining how that documentation can be accessed
- Publicly known vulnerabilities against the platform (part), providing a vulnerability review

Every rationale is based on a self-assessment by the developer that shall be validated by a third-party platform evaluator. The extent of work performed by the platform evaluator depends on the selected assurance level.

4.8 Catalogue of security features and assurance packages

SESIP includes catalogues of security features as an essential part of the methodology:

- The catalogue of Security Functional Requirements (SFRs) which allows for a consistent specification of platforms (parts) and supports a fair comparison between them.
- The catalogue of Security Process Packages (SPPs) which allows the claim and the application verification of specific secure processes to be implemented by the platform developer, at any phase of the platform life cycle.

These catalogues specify a set of security features that are essential to platforms (parts), and for which there is a shared understanding in the community. These security features are therefore more likely to be accessible to product vendors; they also simplify reusability and contribute to SESIP's formalism (see previous subclauses).

Those sets of security features will evolve over time, following the evolution of IoT security challenges, growing usage of the SESIP methodology, and the security responses provided by connected platforms. Platform (part) vendors may also want to differentiate their offerings by including platform-specific or innovative security features that bring added value to their customers, and it is possible to have such features evaluated following the SESIP methodology.

However, it is also expected that the methodology provides the stability required to build and maintain a coherent certification ecosystem. Therefore, in the case where a feature does not match any of the SFRs/SPPs specified in the SESIP catalogue, a vendor may be allowed by the scheme to add product-specific SFRs/SPPs, as long as they are explicitly identified as such, and are clearly separated from the SFRs/SPPs already in the catalogue.

Regarding Security Assurance Requirements (SARs), vendors are not allowed to claim any additional requirements in a SESIP evaluation. However, a specific certification scheme may include limited refinement of the SARs in the SESIP levels.

4.9 SESIP profiles and mappings

4.9.1 General

SESIP profiles and SESIP Mappings are documents intended to support the usability of the SESIP methodology in the field.

4.9.2 SESIP profiles

The SESIP methodology allows the specification of security profiles generic to a type of platform (part); e.g. MCU/MPU, cryptographic library, communication library, etc. These are called SESIP profiles: a SESIP profile document is a generic SESIP Security Target defining the SESIP requirements in terms of security features (see Clause 5) and evaluation activities (see Clause 7) to be addressed during the evaluation of a platform (part) of the type targeted by the profile.

SESIP profiles are intended to be written by entities or groups needing security evaluations for a specific market and/or playing a role in such evaluations. This allows them to ensure that any platform (parts) evaluated against a SESIP profile implements an expected minimum set of features, reaching a targeted security level; also, as the same set of features is assessed based on the same evaluation activities, this allows comparability between evaluations of different evaluated platform (parts).

Thanks to the accessibility of SESIP requirements (see 4.6), stakeholders without specific expertise in evaluation are able to write and understand a SESIP profile.

Note also that profiles allow the consideration of risk and threat analysis for specific type of products; such risk analysis would have previously been established by the relevant actors (e.g. via dedicated working groups of industrials). See Annex F for more details.

4.9.3 SESIP mappings

As specified in 4.5.1, SESIP is intended to work with existing standards, providing them with a framework for the evaluation of their requirements. Indeed, SESIP specifies security requirements (see Clauses 5 and 7) that can be claimed by developers to be addressed in SESIP evaluations. Those requirements have been expressed at a level allowing the coverage of main features of IoT platforms. As a result, requirements of existing standards can easily be mapped to SESIP requirements.

Along with such mapping, the results of the SESIP evaluation can be reused as a demonstration of compliance to an IoT standard, confirming that the IoT standard's security features have been correctly and efficiently implemented.

SESIP Mappings are intended to be written by entities or groups who would like to use SESIP evaluation results as the basis for the demonstration of compliance to a standard. The benefit of the SESIP evaluation in this context is the reuse of an initial SESIP evaluation to demonstrate compliance with several standards, each standard being associated with a dedicated mapping.

5 Security Functional Requirements (SFRs)

5.1 General

Connected platforms need to provide developers and evaluators of IoT applications with functionality to build secure products and allow efficient verification of the accurate use of these secure platforms by users, developers, and evaluators/certifiers. Such needs are expressed by Security Functional Requirements (SFRs) which are generic security features to be implemented by connected platforms.

This clause provides a catalogue of SFRs that have been specified by a group of IoT actors based on their own expertise and on a continuous monitoring of security requirements related to platform layers coming from the field (e.g. through IoT standards) and for which a SESIP evaluation can produce reusable assessment outputs.

The developers shall use this catalogue to select the SFRs to be claimed in the Security Target to specify the security mechanisms implemented by their platform (part) and to be part of the security evaluation scope. The text of the SFRs cannot be modified; however, they can be refined and/or specified more in details through refinements¹ or application notes. This is in particular useful in case of SESIP profile (see 4.9) and/or when mapping to IoT standards requirements.

This catalogue of SFRs can be extended by proprietary SFRs in Security Target and SESIP profiles when none in the catalogue can be used to claim a specific security feature to be part of the evaluation.

The SFRs in this catalogue are specified with the wording of the requirement, the “value”, and the “considerations”.

- The wording of the requirement specifies what shall be implemented; in some requirements, the text includes fields within angle brackets (< >) to be filled with the specificities of platforms (parts).
- The “value” section specifies the value added by a platform providing this functionality, to the users, evaluators, and developers of composite IoT products and platforms.
- The “considerations” section specifies what aspects should be considered in the evaluation and certification of this SFR.

5.2 Identification and attestation of platforms and applications

5.2.1 General

Identification and attestation of applications, platforms, and platform parts allow customers, developers, and evaluators to verify that they have the evaluated product. More complex attestation allows for a wider scope of what is attested, and higher assurance on that which is attested. The SFRs in this subclause are to be used to express identification and attestation requirements.

¹ Refinement rules are as in ISO/IEC 15408-1:2009, 8.1.4.

5.2.2 Verification of platform identity

5.2.2.1 Requirement

5.2.2.1.1 General

The platform provides a unique identification of the platform, including all its parts and their versions.

5.2.2.1.2 Value

Users can verify that they have a secure product only if they can obtain the identifications of the product parts (application and connected platform). The platform is a crucial building block of that identification process.

Evaluators and developers of composites need to be able to verify the identity, too (but might require attestation for higher assurance; see 5.2.3).

5.2.2.1.3 Considerations

The ST specifies the platform, thereby defining all its parts.

Developers of platform (parts) are required to keep the identification (globally) unique: All products from that developer that identify in the way the evaluated product is identified shall be the evaluated product. This functional requirement is mandatory for all platforms, to ensure that customers can verify that they have the correct platform. In case of composition, the platform identification shall also cover the integrated parts.

5.2.3 Verification of platform instance identity

5.2.3.1 Requirement

The platform provides a unique identification of that specific instantiation of the platform, including all its parts.

5.2.3.2 Value

Developers of composites potentially need a unique identifier for a specific instantiation of a platform.

5.2.3.3 Considerations

Developers of platform (parts) are required to keep the individual instantiation identification (globally) unique in combination with the “Verification of platform identity”.

A typical example of platform identity is the platform serial number.

5.2.4 Attestation of platform genuineness

5.2.4.1 Requirement

The platform provides an attestation of the “Verification of platform identity” < and “Verification of platform instance identity” > , in a way that ensures that the platform cannot be cloned or changed without detection.

5.2.4.2 Value

Users, developers, and evaluators can verify that they have the genuine (secure) product, not an insecure/incomplete clone. This gives more assurance to those parties and makes the genuine secure product more distinctive and valuable. Together with the “Verification of platform instance identity” SFR, authentication of this platform can be shown.

Users of the platform certificates issued against this standard, such as dedicated schemes for products, can use this attestation as their way of identifying the genuine product.

5.2.4.3 Considerations

The genuine identification is used to ensure that a given platform is a genuine instantiation of the platform and not a product posturing as one (i.e. a clone), hence the mechanism and its keys are valuable to an attacker seeking to clone the platform or otherwise misuse the developer's brand and reputation. As the developer is potentially held accountable for clones identifying as the genuine platform, or at least suffers reputation damage for them, such an attacker goal shall be considered as part of the evaluation.

5.2.5 Secure initialization of platform

5.2.5.1 Requirement

The platform ensures its integrity and authenticity during platform initialization. If the platform integrity or authenticity cannot be ensured, the platform will go to < list of controlled states > .

5.2.5.2 Value

Users, developers, and evaluators can trust that the platform verified its integrity and authenticity at start-up, hence an operational product is running on a secure platform.

5.2.5.3 Considerations

A platform detecting a breach of integrity or authenticity may offer "Factory reset of platform" and/or "Secure update of platform" functionality to recover a given product, or "Decommission of platform" functionality to neutralize the product.

5.2.6 Attestation of platform state

5.2.6.1 Requirement

The platform provides an attestation of the state of the platform, such that it can be determined that the platform is in a known state.

5.2.6.2 Value

Users, evaluators, and developers of composites can verify that the platform is in a known state that was part of the evaluation.

5.2.6.3 Consideration

Implies that "Attestation of platform genuineness" and "Secure initialization of platform" are also implemented.

5.2.7 Attestation of application genuineness

5.2.7.1 Requirement

The platform provides an attestation of the application, in a way that ensures that the application has not been cloned or changed without detection.

5.2.7.2 Value

Users can determine that they have the genuine (secure) product, not an insecure/incomplete clone. This gives more assurance to the user and makes the genuine (secure) product more distinctive and valuable.

5.2.7.3 Considerations

Implies that “Attestation of platform genuineness” and “Secure initialization of platform” are also implemented.

5.2.8 Attestation of application state

5.2.8.1 Requirement

The platform provides an attestation of the state of the application.

5.2.8.2 Value

Users, evaluators, and developers of composites can verify that the application is in a specific state. This specific state could then be compared to a secure state that was part of the evaluation.

5.2.8.3 Considerations

The attested state of the application is limited to information that is available to the platform, such as the application’s static code and configuration, and other platform-managed information, e.g. an application life cycle state.

Implies that “Attestation of application genuineness” and “Attestation of platform state” are also implemented.

5.3 Product Life Cycle: Factory reset / Install / Update / Decommission

5.3.1 General

The platform shall always maintain the Security Functional Requirements claimed, including the boot and shutdown stages.

The SFRs in this subclause are to be used to express other common life cycle steps such as secure installation, update, and decommission of the platform and application.

The confidentiality of the platform or application may be important, for example to protect intellectual property rights or because this confidentiality has been assumed during a vulnerability analysis. As these life cycle steps may happen in the field, platform and application confidentiality needs to be maintained.

5.3.2 Factory reset of platform

5.3.2.1 Requirement

The platform can be reset to the state in which it exists when the product embedding the platform is delivered to the user, before any personal user data, user credentials, or user configuration is present on the platform.

5.3.2.2 Value

The user invokes this functionality to support the product built on the platform to return to its initial factory state and restart the life-cycle again from this point; this can be used for reselling purposes.

The user’s data are guaranteed to be destroyed, independently of the application running on the platform.

5.3.2.3 Considerations

The platform shall still be functional after the factory reset, allowing the user to re-initialize the product embedding the platform.

This reset shall destroy all data (including the data of the application) received after user delivery, such that none of this data can be compromised even when the product is also physically accessible to an attacker.

This functionality still allows storage of platform instance unique data, such as data needed for “Attestation of platform genuineness” and “Attestation of platform state”, allowing the platform and product to continue to be operational.

This functionality is not intended to counter attacks where an attacker has temporary physical access and then returns it to the user; those are countered by “Physical attacker resistance”.

If the application is (partially) defective, it shall still be possible to factory reset the product and then throw the product away with the (user) data no longer recoverable in the product. Platforms with functionality from “Secure encrypted storage”, “Residual information purging”, or “Cryptographic keyStore” typically will be able to fulfil this requirement more easily.

The destruction method shall be appropriate for the persistent memory technology and attack potential. See also “Residual information purging”.

5.3.3 Secure install of application

5.3.3.1 Requirement

The application can be installed in the field such that the <confidentiality>, integrity and authenticity of the application is maintained.

5.3.3.2 Value

An application may be installed in the field, including at the final end user and at unsecured product production sites. Composite developers and evaluators can be ensured that the application is not modified (or optionally: disclosed) during this installation.

5.3.3.3 Consideration

The installation mechanism shall ensure that an application is compatible with the underlying platform in its current version before installing the application.

A platform offering this SFR should consider also offering “Secure update of application” as a complement to the “Secure update of platform”, allowing the product vendor to provide flaw remediation procedures that cover the entire product.

5.3.4 Secure update of platform

5.3.4.1 Requirement

The platform can be updated to a newer version in the field such that the < confidentiality, > integrity and authenticity of the platform is maintained.

5.3.4.2 Value

Addressing security flaws, functional bugs, or improvements may require an update of the platform in the field. Composite developers and evaluators can be ensured that the update mechanism itself will not enable an attack.

5.3.4.3 Considerations

Note that absence of this functionality shall be explained in the ST under ALC_FLR.SESIP.

The update mechanism shall counter downgrade attacks (“updating” to an older, potentially more vulnerable version). What is an “older” or “newer” version shall be specified in ALC_FLR.SESIP.

Note also that updates of the platform shall have a (globally) unique identifier as per “Verification of platform identity” and may require their own (re-)evaluation and (re-)certification.

5.3.5 Secure update of application

5.3.5.1 Requirement

The application can be updated to a newer version in the field such that the < confidentiality, > integrity and authenticity of the application is maintained.

5.3.5.2 Value

Addressing security flaws, functional bugs, or improvements may require an update of the application in the field. Composite developers and evaluators can be ensured that the application is not disclosed or inappropriately modified during this update.

5.3.5.3 Consideration

The update mechanism shall counter downgrade attacks (“updating” to an older, potentially more vulnerable version). What is an “older” or “newer” version shall be specified in ALC_FLR.SESIP.

The update mechanism shall ensure, before updating the application, that the new version of the application is compatible with the underlying platform in its current version.

A platform offering this may consider also offering “Secure install of application”.

5.3.6 Secure uninstall of application

5.3.6.1 Requirement

The application can be uninstalled in the field such that all application data < with the exception of < list of objects not destroyed > > is destroyed.

5.3.6.2 Value

The user invokes this functionality prior to disposing of the application instance or otherwise potentially allowing an attacker physical access to the application instance. As all application data are destroyed, the application user’s data are also destroyed.

5.3.6.3 Considerations

The uninstallation mechanism shall counter downgrade attacks (uninstall the newer application, “fresh install” to an older, potentially more vulnerable version). What is an “older” or “newer” version shall be specified in ALC_FLR.SESIP.

The secure uninstall shall destroy all data of the application received after installation of the application. If the application developer has marked an application object as exempt from destruction during uninstallation (such as information allowing detection of a downgrade attack), then the uninstall operation shall destroy all application data not explicitly marked as exempt from uninstallation. Destruction shall be such that application data cannot be compromised even when the product is physically accessible to an attacker.

If the application is (partially) defective, it shall still be possible to uninstall the application and then throw the product away with the application (user) data no longer recoverable in the product. Platforms with functionality from “Secure encrypted storage”, “Residual information purging”, or “Cryptographic keyStore” will typically be able to fulfil this requirement more easily.

The destruction method shall be appropriate for the persistent memory technology and attack potential. See also “Residual information purging”.

5.3.7 Decommission of platform

5.3.7.1 Requirement

The platform can be decommissioned.

5.3.7.2 Value

The user invokes this functionality to reach the end-of-life state of the platform, with no way back to operational state. As all data are destroyed, the user's data are also destroyed.

5.3.7.3 Considerations

The platform shall not be functional after the decommissioning.

The decommissioning shall destroy all data (including the data of the application) received after production.

5.3.8 Field return of platform

5.3.8.1 Requirement

The platform can be returned to the vendor without user data.

5.3.8.2 Value

The user invokes this functionality prior to returning the product instance to its provider in case investigation is needed e.g. malfunction observed in the field. As the user's data are destroyed, the user can be ensured that not even the vendor can access personal information.

5.3.8.3 Considerations

The field return shall destroy all data (including the data of the application) received after user delivery, such that none of this data can be compromised even when the product is physically accessible to an attacker or the vendor.

This functionality still allows storage of platform instance unique data, such as data needed for "Attestation of platform genuineness" and "Attestation of platform state", allowing the platform and product to continue to be operational.

This functionality is not intended to counter attacks where an attacker has temporary physical access and then returns it to the user; those are countered by "Physical attacker resistance".

5.4 Secure communication

5.4.1 General

Typically, a product will use secure channels to communicate with a server, another connected product, or a cloud service. The SFRs in this subclause can be used to describe the platform providing such functionality to the application.

Note that the secure communication SFRs should be used when the platform provides the secure channels; if the platform provides only the cryptographic services that will be used to implement the secure channels, then the SFRs specified in 5.6, Cryptographic functionality, can be used.

5.4.2 Secure communication support

5.4.2.1 Requirement

The platform provides one or more secure communication channel(s).

The secure communication channel authenticates < list of end points > and protects against < list of attacks including disclosure, modification, replay, erasure > of messages between the end points, using < list of protocols and measures > .

5.4.2.2 Value

The product developer can use the secure communication channels.

The composite evaluator/certifier knows that if the product developer uses the functionality, it is secure in the above manner.

The user, relying on the product developer and composite certifier, is ensured that secure channels are used.

5.4.2.3 Considerations

Complete the variable parts of this SFR as follows:

- List of end points: The list of end points to be protected using the protocols and measures listed hereafter. End points can be identified by their quality, by the interface they are connected to, or by a more general category, e.g. all local end points. If different protocols and measures are used for different end points, defining different secure communication channels, then the SFR shall be iterated.
- List of attacks: The list of security issues to avoid, typically including disclosure, modification, replay, and erasure.
- List of protocols and measures: The list of protocols and other measures used to protect the communication channel, e.g. TLS 1.2 with TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8, or IPSEC with It is important to provide enough information to properly characterize the quality of the algorithms used.

If multiple different secure communication channels are provided, this SFR should be iterated.

This SFR could be sensitive to misleading claims; the fully authorized SESIP CB and evaluators shall ensure this is not misleading.

5.4.3 Secure communication enforcement

5.4.3.1 Requirement

The platform ensures that the communication with < list of end points > can only be done over the secure communication channel(s) supported by the platform using < list of protocols and measures > .

5.4.3.2 Value

The user and composite evaluator/certifier can trust that the product is securely communicating (typically with cloud services).

5.4.3.3 Considerations

This SFR requires that at least one iteration of the “Secure communication support” SFR is claimed.

Complete the variable parts of this SFR as follows:

- List of end points: The list of end points identifies the secure channels the SFR applies to, as they have been specified in the corresponding “Secure communication support” SFRs.
- List of protocols and measures: The list of protocols and other measures used for the secure channel implementation.

This SFR implies that the platform handles all the unsecured communication layers under the secure communication channel (TCP/IP, DHCP, DNS, BT, etc.) and that the application does not have direct access to the communication layers underlying that secure channel.

This SFR could be sensitive to misleading claims; the fully authorized SESIP CB and evaluators shall ensure this is not misleading.

5.5 Extra attacker resistance

5.5.1 General

In our regular attacker model, attackers have only logical (network) access to the product, application, and platform during the exploitation phase of the attack, and therefore:

- do not have physical access to the specific product instance attacked, and
- are unable to run their own hostile code on the platform.

When this attacker model is not valid, adding SFRs from this subclause allows expression of resistance against physical and software attackers.

Note that during the identification phase, the attacker is assumed to have physical access to his platform instance when preparing his attack.

5.5.2 Limited physical attacker resistance

5.5.2.1 Requirement

The platform detects or prevents attacks by an attacker with physical access before the attacker compromises < list of Security Functional Requirements > .

5.5.2.2 Value

For situations where a physical attacker is in scope for a limited set of functionalities such as valuable assets warranting a physical attack, but not for all functionality. The attacker is not limited in his physical attack; the attacker is limited in the target of his attack (the limited set of SFRs).

5.5.2.3 Considerations

All attacks enabled by physical access within the attack potential need to be considered.

With this SFR, local non-networked interfaces such as a USB port or MicroSD card reader shall now also be considered in the vulnerability analysis as accessible to the attacker (such as an “evil maid”).

Attackers (mis-)using production and debug functionality such as JTAG and ISD functionality would typically be countered by disabling this functionality prior to delivery to the customer. Invasive attacks such as physical tampering or perturbation would typically be countered by detection and decommissioning the product before the detected attack succeeds. Non-invasive attacks such as side channel attacks would typically be countered by not leaking secret information such as timing, power, and EM emissions, via side channels.

The threat of replacement of the product is not covered by this SFR; it can be countered with SFRs “Verification of platform instance identity” together with “Attestation of platform genuineness”, allowing an application or user to detect replacement.

5.5.3 Physical attacker resistance

5.5.3.1 Requirement

The platform detects or prevents attacks by an attacker with physical access before the attacker compromises any of the other security functional requirements.

5.5.3.2 Value

For situations where a physical attacker is in scope, such as products that are typically used outside a secured area, or when the products store highly valuable assets warranting a physical attack.

5.5.3.3 Considerations

All attacks enabled by physical access within the attack potential need to be considered.

With this SFR, local non-networked interfaces such as a USB port or MicroSD card reader shall now also be considered in the vulnerability analysis as accessible to the attacker (such as an “evil maid”).

Attackers (mis-)using production and debug functionality such as JTAG and ISD functionality would typically be countered by disabling this functionality prior to delivery to the customer. Invasive attacks such as physical tampering or perturbation would typically be countered by detection and decommissioning the product before the detected attack succeeds. Non-invasive attacks such as side channel attacks would typically be countered by not leaking secret information, such as timing, power, and EM emissions, via side channels.

The threat of replacement of the product is not covered by this SFR; it can be countered with SFRs “Verification of platform instance identity” together with “Attestation of platform genuineness”, allowing an application or user to detect replacement.

5.5.4 Software attacker resistance: Isolation of platform

5.5.4.1 Requirement

The platform provides isolation between the application and itself, such that an attacker able to run code as an application on the platform cannot compromise the other security functional requirements.

5.5.4.2 Value

For situations where an attacker may be able to load his own code on the platform, or the attacker might subvert any part of the application.

5.5.4.3 Considerations

This would typically require at least an OS with kernel-user mode separation.

5.5.5 Software attacker resistance: Isolation of platform parts

5.5.5.1 Requirement

The platform provides isolation between platform parts, such that an attacker able to run code in < list of vulnerable platform parts > can compromise neither the < confidentiality and > integrity of < list of protected platform parts > nor the provision of any other Security Functional Requirements.

5.5.5.2 Value

For situations where an attacker may be able to load his own code on inner parts of the platform.

The platform developer can separate the critical assets in different parts of the platform, and thus safeguard them from compromises of other parts of the platform.

5.5.5.3 Considerations

Complete the variable parts of this SFR as follows:

- The list of vulnerable platform parts specifies the parts of the platform that are considered to be potentially compromised by an attacker and used to attack other parts of the platform.
- The list of protected platform parts specifies the parts of the platform that are considered to be potentially targeted by attackers running code in other compromised parts, and that are protected against such attacks.

A platform offering this functionality would typically also claim “Software attacker resistance: Isolation of platform”.

This would typically require a micro-kernel or a hardware-based isolation technology.

5.5.6 Software attacker resistance: Isolation of application parts

5.5.6.1 Requirement

The platform provides isolation between parts of the application, such that an attacker able to run code as one of the < list of application parts > cannot compromise the < confidentiality and > integrity of the other application parts.

5.5.6.2 Value

For situations where the product developer wants to separate the critical assets in its own application part (process, executable, etc.), and thus safeguard those assets from compromises by other parts of the application code that are too complex to be shown to be secure.

5.5.6.3 Considerations

The variable part of this SFR, list of application parts, shall list the part types that may compose the application (modules, processes, applets), and between which some isolation is to be provided.

This would typically require an OS with application-application memory separation or an interpreter-based platform with similar access rules, such as Java Card.

5.6 Cryptographic functionality

5.6.1 General

These are common cryptographic functions that a platform can provide that are useful to the developer of a composite or to a product user.

All of the claimed parts of the specification need to be fully implemented, so precise references are encouraged. Evaluators and certifiers shall verify all aspects of the parts of standards referenced.

5.6.2 Cryptographic operation

5.6.2.1 Requirement

The platform provides the < list of cryptographic operations > functionality with < list of algorithms > as specified in <specification> for key lengths < list of key lengths > and modes < list of modes > .

5.6.2.2 Value

Evaluators and developers of composites can be ensured of standard-compliant cryptographic functions.

5.6.2.3 Considerations

Complete the variable parts of this SFR as follows:

- The list of cryptographic operations defines the operations that can be considered, typically selected from encryption, decryption, hashing, signing, and signature verification.
- The list of algorithms provides a reference to the cryptographic algorithms used.
- The specification references the standard in which the operation is specified (including a section or similar information if relevant).
- The list of key lengths specifies the key lengths that are supported for that cryptographic operation.
- The list of modes specifies the operational modes that are supported for that cryptographic operation.

A typical example of a fully defined SFR would be: The platform provides the application with encryption and decryption functionality as specified in NIST FIPS 197 (AES) with 256-bit keys in GCM mode.

This SFR should be iterated if more than one algorithm is provided. For the sake of clarity, it may be preferable to use a table as Table 5-1 to specify the supported parameters:

Table 5-1 — Cryptographic operations example

Operations	Algorithm	Specification	Key lengths	Modes
Encryption, decryption	AES	NIST FIPS 197	256	GCM

The cryptographic operation shall keep the confidentiality of the secret keys against the attacker. Even with the standard attacker model, timing and padding oracle attacks shall be considered if they are within the attack potential.

Note that this SFR is for the cryptographic functionality available to the application. This SFR should not be used to only restate the crypto functionality claimed in other SFRs (such as “Secure encrypted storage”, “Secure communication support”, “Secure communication enforcement”) unless that functionality is separately made available to the application.

5.6.3 Cryptographic key generation

5.6.3.1 Requirement

The platform provides a way to generate cryptographic keys for use in <list of cryptographic algorithms> as specified in <specification> for key lengths <list of key lengths> .

5.6.3.2 Value

Evaluators and developers of composites can be ensured of standard-compliant cryptographic key generation.

5.6.3.3 Considerations

Complete the variable parts of this SFR as follows:

- The list of cryptographic algorithms provides a reference to the algorithms used.
- The specification references the standard in which the operation is specified (including a section or similar information if relevant).
- The list of key lengths specifies the key lengths that are supported for that cryptographic operation.