



HARMONISED EUROPEAN STANDARD

Cyber Security (CYBER); CRA; Cybersecurity requirements for Operating Systems (OS)

Exercising one of the Principles of international standardization - Openness - and taking it to a different level, ETSI is piloting informal public consultations of the vertical standards in support of the Cyber Resilience Act at a much earlier stage than it is usual in the standardization world. In this context, please keep in mind that the standard draft herein is an INTERIM DRAFT, which expectably will be subject to substantial changes before its target publication date in the second semester of 2026.

Disclaimer

This **INTERIM DRAFT** document is provided for information and is for future development work within the ETSI Technical Committee CYBER EUSR Working Group (CYBER-EUSR) only. ETSI and its Members accept no liability for any further use/implementation of this Specification. Approved and published specifications and reports shall be obtained exclusively via the ETSI Documentation Service at <http://www.etsi.org/standards-search>

Commenting guidelines

Your comments to improve this early draft are very welcomed. To ensure the effectiveness of your contribution, please make sure to include the following elements in your comment:

1. Clear identification of the section of the draft your comment is referring to.
2. Objective proposal to delete content, add content or substitute content.
3. Concrete contribution (exact content you suggest including in the draft as an addition to, or in substitution of, existing content).
4. Brief rationale to justify the proposal (e.g. why the suggested content should be added an/or the existing content should be deleted or substituted). Please note that comments without concrete contributions will be noted but not acted upon by ETSI CYBER-EUSR. We trust and value your expertise and therefore expect you to take this opportunity to proactively contribute to solving the issues you find while analysing this early draft.

How to comment

To comment or provide feedback on the present interim draft please visit: [STAN4CRA / EN 304 626 Operating Systems - GitLab](#) and submit your comments as “issues” following the guidelines provided on this site.

Alternatively, you may also send your comments to: cybersupport@etsi.org using the “[ETSI Commenting Format for Open Consultation.xlsx](#)” available in <https://docbox.etsi.org/CYBER/EUSR/Open>

Reference

DEN/CYBER-EUS-0012

Keywords

CRA, Cybersecurity, OS>

ETSI

650 Route des Lucioles

F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - APE 7112B

Association à but non lucratif enregistrée à la Sous-préfecture de Grasse (06) N° w061004871

Important notice

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the prevailing version of an ETSI deliverable is the one made publicly available in PDF format on [ETSI deliver](#) repository.

Users should be aware that the present document may be revised or have its status changed, this information is available in the [Milestones listing](#).

If you find errors in the present document, please send your comments to the relevant service listed under [Committee Support Staff](#).

If you find a security vulnerability in the present document, please report it through our [Coordinated Vulnerability Disclosure \(CVD\)](#) program.

Notice of disclaimer & limitation of liability

The information provided in the present deliverable is directed solely to professionals who have the appropriate degree of experience to understand and interpret its content in accordance with generally accepted engineering or other professional standard and applicable regulations.

No recommendation as to products and services or vendors is made or should be implied.

No representation or warranty is made that this deliverable is technically accurate or sufficient or conforms to any law and/or governmental rule and/or regulation and further, no representation or warranty is made of merchantability or fitness for any particular purpose or against infringement of intellectual property rights.

In no event shall ETSI be held liable for loss of profits or any other incidental or consequential damages.

Any software contained in this deliverable is provided "AS IS" with no warranties, express or implied, including but not limited to, the warranties of merchantability, fitness for a particular purpose and non-infringement of intellectual property rights and ETSI shall not be held liable in any event for any damages whatsoever (including, without limitation, damages for loss of profits, business interruption, loss of information, or any other pecuniary loss) arising out of or related to the use of or inability to use the software.

Copyright Notification

No part may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm except as authorized by written permission of ETSI. The content of the PDF version shall not be modified without the written authorization of ETSI. The copyright and the foregoing restriction extend to reproduction in all media.

© ETSI 2025. All rights reserved.

Contents

HARMONISED EUROPEAN STANDARD

Intellectual Property Rights

Essential patents

Trademarks

Foreword

Modal verbs terminology

Executive summary

Introduction

1 Scope

1.1 General

1.2 Products in scope

1.2.1 General

1.2.2 Components of operating systems that are in scope

1.3 Products not in scope

2 References

2.1 Normative references

2.2 Informative references

3 Definition of terms, symbols and abbreviations

3.1 Terms

3.2 Symbols

3.3 Abbreviations

4 Product context

4.1 Intended purpose and reasonably foreseeable use

4.2 Essential functions

4.3 Product architecture

4.3.1 Overview

4.3.2 Operating system security functions

4.3.3 High-level operating system architectures

4.3.4 Access control mechanisms

4.3.5 Resource management

4.3.6 Scheduling

4.4 Operational Environment

4.5 Distribution of security functions

4.5.1 General

4.5.2 Security functions provided outside the product

4.5.3 Security functions provided to other components

4.6 Users

4.7 Use Cases

4.7.1 UC-LR: Operating system for learning and research

4.7.2 UC-IoT-1: Non-internet-connected device such as a bluetooth speaker

4.7.3 UC-IoT-2: Internet-enabled power switch

4.7.4 UC-IoT-3: Internet-connected “smart home” device

4.7.5 UC-RO-1: Consumer-grade home wireless router

4.7.6 UC-OT-1: Business-grade remote door locking system

4.7.7 UC-MOB-1: Personal mobile device

4.7.8 UC-WE-1: Wearable health tracker

4.7.9 UC-PC-1: Personal computer in a fixed and generally safe location

4.7.10 UC-PC-2: Enterprise workstation in a fixed and generally safe location

4.7.11 UC-LA-1: Personal laptop

4.7.12 UC-LA-2: Enterprise laptop

4.7.13 UC-PS-1: Personal server

4.7.14 UC-SE-1: Enterprise server in a datacenter with no user accounts

4.7.15 UC-SE-2: Enterprise server in a datacenter with only trusted user accounts

4.7.16 UC-SE-3: Enterprise server in a datacenter hosting many untrusted user accounts

5 Requirements specifications

5.1 Notes on the structure of the Requirements

5.1.1 Necessity of Requirements

5.1.2 Types of Technical Requirements

5.1.3 Assumptions Regarding Requirements

- [5.1.3.1 Testability](#)
- [5.1.3.2 Source Code](#)
- [5.1.3.3 Mitigations](#)
- [5.1.3.4 Risk Transferral](#)
- [5.2 Technical security requirements specifications](#)
- [5.2.1 General](#)
- [5.2.2 TR-NKEV: No known exploitable vulnerabilities at first use](#)
- [5.2.2.1 Requirement](#)
- [5.2.2.2 MI-KEVD: Documentation for secure update before or during first use](#)
- [5.2.2.3 MI-KEVA: Automatic secure update before or during first use](#)
- [5.2.2.4 MI-KEVM: Documentation of mitigation of known exploitable vulnerabilities](#)
- [5.2.2.5 MI-KEVT: Testing for known exploitable vulnerabilities](#)
- [5.2.2.6 MI-SCAN: No easily scannable known exploitable vulnerabilities](#)
- [5.2.3 TR-SSDD: Secure design and development](#)
- [5.2.3.1 Requirement](#)
- [5.2.3.2 MI-SSCA: Static source code analysis for memory errors](#)
- [5.2.3.3 MI-FZ95: Runtime code coverage checking with memory access error detection](#)
- [5.2.3.4 MI-IMSL: Implement in a memory-safe language](#)
- [5.2.3.5 MI-BTIN: Boundary testing of inputs that may cause memory errors](#)
- [5.2.3.6 MI-SCFS: Secure compilation flags](#)
- [5.2.4 TR-MISO: Prevent local unauthorized access of memory-addressable security-relevant data](#)
- [5.2.4.1 Requirement](#)
- [5.2.4.2 MI-MMAC: Memory access control](#)
- [5.2.4.3 MI-CCON: Prevent creation of more than one user account](#)
- [5.2.4.4 MI-UCON: Prevent concurrent user account usage](#)
- [5.2.4.5 MI-PMSC: Prevent memory leaks through microarchitectural side channels in provided executables](#)
- [5.2.4.6 MI-TRMD Transfer risk of microarchitectural side channel data leaks to user](#)
- [5.2.5 TR-MSAF: Mitigate memory safety errors](#)
- [5.2.5.1 Requirement](#)
- [5.2.5.2 Default Preparation, Verdict, and Evidence](#)
- [5.2.5.3 MI-MSAF-1: Stack exhaustion detection](#)
- [5.2.5.4 MI-MSAF-2: Stack linear buffer overflow detection](#)
- [5.2.5.5 MI-MSAF-3: Array bounds checking](#)
- [5.2.5.6 MI-MSAF-4: Heap linear buffer overflow detection](#)
- [5.2.5.7 MI-MSAF-5: Heap use-after-free access prevention](#)
- [5.2.5.8 MI-MSAF-6: Heap free checking](#)
- [5.2.6 TR-LMII: Limit incident impact](#)
- [5.2.6.1 Requirement](#)
- [5.2.6.2 Default Preparation, Verdict, and Evidence](#)
- [5.2.6.3 MI-MZRO-1: Stack memory zeroing](#)
- [5.2.6.4 MI-MZRO-2: Heap memory zeroing](#)
- [5.2.6.5 MI-MRWX-1: Prevent writes to executable and read-only data memory](#)
- [5.2.6.6 MI-MRWX-2: Prevent execution of non-kernel code memory](#)
- [5.2.6.7 MI-ASLR Address space layout randomization](#)
- [5.2.6.8 MI-MRCO: Mitigate reference counter overflow](#)
- [5.2.6.9 MI-NKAM: Prevent unintentional kernel access to userspace memory](#)
- [5.2.6.10 MI-PLLC: Prevent linked list corruption](#)
- [5.2.6.11 MI-CFIN: Control flow integrity](#)
- [5.2.6.12 MI-MPMT: Memory protection using memory tagging](#)
- [5.2.7 TR-MINI: Minimize impact on other devices and services](#)
- [5.2.7.1 Requirement](#)
- [5.2.7.2 MI-MDOC: Document transfer of risk of minimizing impact to operating environment](#)
- [5.2.7.3 MI-MNET: Minimize negative impact of network transmission](#)
- [5.2.7.4 MI-MAMP: Minimize negative impact of network traffic amplification](#)
- [5.2.8 TR-SDEF: Secure by default configuration](#)
- [5.2.8.1 Requirement](#)
- [5.2.8.2 MI-ADEF: Authorization required by default to access security-relevant assets](#)
- [5.2.8.3 MI-PDDI-1: Document how to protect access to debug and management interfaces](#)
- [5.2.8.4 MI-PDDI-2: Protect or disable physical access to debug and management interfaces](#)
- [5.2.8.4 MI-PDDI-3: Protect or disable local software access to debug and management interfaces](#)
- [5.2.8.5 MI-PDDI-4: Protect or disable network access to debug or management interfaces](#)
- [5.2.9 TR-SCUD: Secure updates](#)

- [5.2.9.1 Requirement](#)
- [5.2.9.2 MI-SCHL: Low security updates provided by operational environment](#)
- [5.2.9.3 MI-SCHM: Medium security updates provided by operational environment](#)
- [5.2.9.4 MI-SCHH: High security updates provided by operational environment](#)
- [5.2.9.5 TODO](#)
- [5.2.7 TR-AUTH: Authentication and access control](#)
- [5.2.7 TR-CDST: Confidentiality of data stored on the product](#)
 - [5.2.7.1 Requirement](#)
 - [5.2.7.2 MI-CDST: Protect confidentiality of data stored on the product](#)
- [5.2.8 TR-CDTX: Confidentiality of data transmitted by product](#)
 - [5.2.8.1 Requirement](#)
 - [5.2.8.2 MI-CDTX: Protect confidentiality of data transmitted by product](#)
 - [5.2.8.3 MI-DOCC: Document transfer of risk of confidentiality of data transmitted by product](#)
- [5.2.9 TR-CRYP: Encryption](#)
- [5.2.10 TR-IDST: Integrity of data stored on the product](#)
 - [5.2.10.1 Requirement](#)
 - [5.2.10.2 MI-IDST: Protect integrity of data stored on the product](#)
 - [5.2.10.3 MI-DCST: Detect corruption of data stored](#)
- [5.2.11 TR-IDTX: Integrity of data transmitted by the product](#)
 - [5.2.11.1 Requirement](#)
 - [5.2.11.2 MI-DCTX: Detect corruption of data transmitted by the product](#)
- [5.2.12 TR-DMIN: Data Minimization](#)
 - [5.2.12.1 Requirement](#)
 - [5.2.12.2 MI-DJST: Document and justify processed data](#)
- [5.2.13 TR-AVAI: Availability](#)
 - [5.2.13.1 Requirement](#)
 - [5.2.13.2 MI-AVNT: Availability of network services](#)
 - [5.2.13.3 MI-WDOG: Watchdog and self-initiated reset](#)
 - [5.2.13.4 MI-FDRP: Fast packet drop](#)
 - [5.2.13.5 MI-LMEM: Limit memory usage](#)
 - [5.2.13.6 MI-FAIR: Fair resource usage and prioritization](#)
 - [5.2.13.7 MI-DOST: Document risk transfer to operational environment for denial of service](#)
- [5.2.14 TR-LMAS: Minimize exposed interfaces](#)
 - [5.2.14.1 Requirement](#)
 - [5.2.14.2 MI-JSTY: Document and justify exposed interfaces](#)
- [5.2.15 TR-LOGG: Logging and monitoring](#)
 - [5.2.15.1 Requirement](#)
 - [5.2.15.2 MI-LOGG: Logging](#)
- [5.2.16 TR-SCDL: Secure deletion](#)
 - [5.2.16.1 Requirement](#)
 - [5.2.16.2 MI-RSET: Secure deletion via reset](#)
 - [5.2.16.3 MI-INST: Secure deletion via reinstallation](#)
 - [5.2.16.4 MI-DELE: Secure deletion via secure deletion function](#)
- [5.2.17 TR-SDTR: Secure data read and transfer](#)
 - [5.2.17.1 Requirement](#)
 - [5.2.17.2 MI-SDRF: Secure data read from product](#)
 - [5.2.17.3 MI-SDTR: Secure data transfer to another product](#)
- [5.2.18 TR-VULH: Vulnerability handling](#)
 - [5.2.18.1 Requirement](#)
 - [5.2.18.2 MI-VULH-1: Vulnerability Handling in the Product](#)
 - [5.2.18.3 MI-VULH-2: Enabling Vulnerability Handling in Integrated Products](#)
- [5.3 Risk Mitigation Sets](#)
 - [5.3.1 General](#)
 - [5.3.2 SP-LR required mitigations](#)
 - [5.3.3 SP-IoT-1 required mitigations](#)
 - [5.3.4 SP-IoT-2 required mitigations](#)
 - [5.3.5 SP-IoT-3 required mitigations](#)
 - [5.3.6 SP-RO-1 required mitigations](#)
 - [5.3.7 SP-OT-1 required mitigations](#)
 - [5.3.8 SP-MOB-1 required mitigations](#)
 - [5.3.9 SP-WE-1 required mitigations](#)
 - [5.3.10 SP-PC-1 required mitigations](#)

- [5.3.11 SP-PC-2 required mitigations](#)
- [5.3.12 SP-LA-1 required mitigations](#)
- [5.3.13 SP-LA-2 required mitigations](#)
- [5.3.14 SP-PS-1 required mitigations](#)
- [5.3.15 SP-SE-1 required mitigations](#)
- [5.3.16 SP-SE-2 required mitigations](#)
- [5.3.17 SP-SE-3 required mitigations](#)

[6 Conformity Assessment](#)

[Annex A \(informative\): Mapping between the present document and CRA requirements](#)

[Annex B \(informative\): Relationship between the present document and any related ETSI standards \(if any\)](#)

[Annex C \(informative\): Risk identification and assessment methodology](#)

[C.1 Assets](#)

[C.1.1 Data](#)

[C.1.2 Product functions](#)

[C.2 Risk factors](#)

[C.2.1 General comments regarding risk factors](#)

[C.2.2 RF-NUSR: Number of User Accounts](#)

[C.2.3 RF-CUSR: User Account Concurrency](#)

[C.2.4 RF-PPII: Potential for Collection of Personally Identifiable Information](#)

[C.2.5 RF-SNDS: Sensitivity of Data Stored](#)

[C.2.6 RF-SNDT: Sensitivity of Data Transmitted](#)

[C.2.7 RF-SENF: Sensitivity of Functions](#)

[C.2.8 RF-PHYS: Physical Access by Threat Actors to the Device](#)

[C.2.9 RF-UEIN: Processing of Untrusted External Inputs](#)

[C.2.10 RF-LOSS: Probability of Loss of the Device](#)

[C.2.11 RF-HWMD: Hardware Modifiability by End Users](#)

[C.2.12 RF-SWMD: Software Modifiability by End Users](#)

[C.2.13 RF-DVCS: Untrusted Peripheral Devices](#)

[C.2.14 RF-TNET: Access to a Public Network](#)

[C.2.15 RF-FNET: Accessed From Untrusted Networks Including a Public Network](#)

[C.2.16 RF-CONF: Configurability](#)

[C.2.17 RF-ADMN: Administration](#)

[C.2.18 RF-SUPP: Support and Foreseeable Updates](#)

[C.3 Assumptions](#)

[C.3.1 AS-PP: Proper platform](#)

[C.3.2 AS-PA: Proper administrator](#)

[C.3.3 AS-LP: Attacker has limited physical access to product](#)

[C.3.4 AS-LR: Attacker has limited resources](#)

[C.4 Threats and risk assessments of threats](#)

[C.4.1 General](#)

[C.4.2 Risk assessment methodology](#)

[C.4.3 TH-UEVU: Unknown exploitable vulnerabilities](#)

[C.4.4 TH-KEVU: Known exploitable vulnerabilities](#)

[C.4.5 TH-UAPP: Unauthorized access to product assets via unprotected physical interfaces in default configuration](#)

[C.4.6 TH-UAPS: Unauthorized access to product assets via unprotected local software access in default](#)

[configuration](#)

[C.4.7 TH-UAPN: Unauthorized access to product assets via unprotected network interfaces in default](#)

[configuration](#)

[C.4.9 TH-UADT: Unauthorized access to confidential data transmitted](#)

[C.4.10 TH-PDOS: Denial of service attack on product functions via user or network access](#)

[C.4.11 TH-DDOS: Denial of service attack on other products via exploitation of vulnerabilities or unauthorized](#)

[use of product functions](#)

[C.4.12 TH-MQSE: Masquerading authorized server](#)

[C.4.13 TH-LEAK: Data leak through side channels](#)

[C.5 Mapping of use cases to risk factors](#)

[C.6 Security profiles and security assurance levels](#)

[C.6.1 General](#)

[C.6.2 Mapping of security profiles to risk factors](#)

[C.6.3 Security assurance levels](#)

[C.6.4 Mapping of security profile to security assurance level](#)

[Annex D \(informative\): Risk evaluation guidance](#)

[D.1 Explanation of Risk Modeling Approach](#)

[D.2 Mapping of risks to requirements](#)

[D.3 Risk acceptance criteria](#)

[D.4 Risks not treated by the requirements](#)

[Annex E \(informative\): Change history](#)

[History](#)

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: "*Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards*", which is available from the ETSI Secretariat. Latest updates are available on the [ETSI IPR online database](#).

Pursuant to the ETSI Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™**, **LTE™** and **5G™** logo are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM®** and the GSM logo are trademarks registered and owned by the GSM Association.

Foreword

This draft Harmonised European Standard (EN) has been produced by ETSI Technical Committee Cyber Working Group for EUSR (CYBER-EUSR), and is now submitted for the combined Public Enquiry and Vote phase of the ETSI Standardisation Request deliverable Approval Procedure (SRdAP).

The present document has been prepared under the Commission's standardisation request C(2025) 618 final to provide one voluntary means of conforming to the requirements of Regulation (EU) No 2024/2847 of the European Parliament and of the Council of 23 October 2024 on horizontal cybersecurity requirements for products with digital elements and amending Regulations (EU) No 168/2013 and (EU) No 2019/1020 and Directive (EU) 2020/1828 (Cyber Resilience Act).

Once the present document is cited in the Official Journal of the European Union under that Regulation, compliance with the normative clauses of the present document given in table A.1 confers, within the limits of the scope of the present document, a presumption of conformity with the corresponding requirements of that Regulation and associated EFTA regulations.

Modal verbs terminology

In the present document “**shall**”, “**shall not**”, “**should**”, “**should not**”, “**may**”, “**need not**”, “**will**”, “**will not**”, “**can**” and “**cannot**” are to be interpreted as described in clause 3.2 of the [ETSI Drafting Rules](#) (Verbal forms for the expression of provisions).

“**must**” and “**must not**” are **NOT** allowed in ETSI deliverables except when used in direct citation.

Executive summary

Introduction

1 Scope

1.1 General

The present document specifies security requirements and related assessment criteria regarding the compliance of Operating Systems with EU Regulation 2024/2847.

Following harmonised standards in the design and manufacture of products may ensure that the products comply with corresponding EU rules.

The use of harmonised standards are voluntary.

1.2 Products in scope

1.2.1 General

Products in scope are products whose core function and intended or reasonably foreseeable use or misuse is as an operating system. Operating systems include software products with digital elements that provide an abstract interface of the underlying hardware and control the execution of software, and that may provide services such as computing resource management and configuration, scheduling, input-output control, managing data, and providing an interface through which applications interact with system resources and peripherals. The underlying hardware may be virtualized to some degree, as when an operating system is running on a hypervisor.

This category includes but is not limited to:

- General purpose operating systems
 - Personal computing operating systems
 - Mobile operating systems
 - Server operating systems
- Special purpose operating systems
 - Real-time operating systems
 - Embedded operating systems
 - Single-purpose operating systems

Many products contain multiple operating systems which can affect the security functions of other operating system(s) in the product. For example, a Baseboard Management Controllers (BMC) contains an operating system that can manage most or all of the hardware managed by the main system operating system. Radiofrequency transmission devices often have an embedded real-time operating system and the ability to read or write to system memory or trigger interrupts.

Some of the operating systems may not always be readily available as separate products and are included as components of another product. Where there may be other specifications that target that product category, it may be more relevant to review the operating system as part of that larger system rather than independently via this standard.

1.2.2 Components of operating systems that are in scope

The scope of this document is limited to the security-relevant parts of the operating system, including components that provide or are capable of modifying or controlling essential security functions of the operating system.

The following non-exhaustive list of types of components are common to many operating systems and, when present, are considered security-relevant:

- **Kernel:** The central component responsible for managing hardware resources and enforcing access controls.

- **Device Drivers:** Software components supplied with the operating system that interact directly with hardware devices.
- **Security Libraries:** Libraries used to provide critical security services, such as encryption, authentication, and authorization.
- **Authentication Services:** Core authentication mechanisms required for operating system functionality.
- **Privileged Processes:** Operating system processes running with elevated privileges or access to sensitive resources.
- **Software Update Mechanisms:** Systems responsible for installing and updating software components supplied with the operating system.
- **Logging and Monitoring:** Functions performed by the operating system that record security-relevant events or monitor system behavior.
- **Configuration Management:** Management of the configuration of security-relevant operating system settings, including provisioning of secure-by-default configuration as appropriate to the product context.

Other components of operating systems often contribute to the essential security of a product and should be given equal consideration, as appropriate for each product's context.

1.3 Products not in scope

The present document does not apply to products that contain an operating system or are part of an operating system if the core purpose of the product is not that of an operating system. However, it may be useful as one part of the process of demonstrating compliance for a product containing or interacting with an operating system.

The present document does not cover functions of the operating system that are not security-relevant.

The present document does not cover other product categories defined in EU Regulation 2024/2847, such as hypervisors, container runtime systems, or boot managers, even where such products provide security-relevant functionality which overlaps that of an operating system.

2 References

2.1 Normative references

The following referenced documents are necessary for the application of the present document.

- [1] prEN 40000-1-1: “Cybersecurity requirements for products with digital elements – Vocabulary”
- [2] prEN 40000-1-2: “Cybersecurity requirements for products with digital elements - Part 1-2: Principles for cyber resilience”
- [3] prEN 40000-1-3: “Cybersecurity requirements for products with digital elements – Vulnerability Handling”

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or nonspecific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long-term validity.

The following referenced documents may be useful in implementing an ETSI deliverable or add to the reader’s understanding but are not required for conformance to the present document.

- [i.1] Regulation (EU) 2024/2847 of the European Parliament and of the Council of 23 October 2024 on horizontal cybersecurity requirements for products with digital elements and amending Regulations (EU) No 168/2013 and (EU) 2019/1020 and Directive (EU) 2020/1828 (Cyber Resilience Act).
- [i.2] C(2025)618 – Standardisation request M/606: Commission Implementing decision of 3.2.2025 on a standardisation request to the European Committee for Standardisation (CEN), the European Committee for Electrotechnical Standardisation (Cenelec) and the European Telecommunications Standards Institute (ETSI) as regards products with digital elements in support of Regulation (EU) 2024/2847 of the European Parliament and of the Council of 23 October 2024 on horizontal cybersecurity requirements for products with digital elements and amending Regulations (EU) No 168/2013 and (EU) 2019/1020 and Directive (EU) 2020/1828 (Cyber Resilience Act).
- [i.3] ETSI EN 303 645: “CYBER; Cyber Security for Consumer Internet of Things: Baseline Requirements”.
- [i.4] ETSI TS 103 701: “Cyber Security (CYBER); Cyber Security for Consumer Internet of Things: Conformance Assessment of Baseline Requirements”.
- [i.5] EN 18031 series: “Common security requirements for radio equipment” (produced by CEN/CENELEC).
- [i.6] CEN/CLC JTC13: “Cybersecurity and Data Protection”.
- [i.7] ISO/IEC 15408: “Information security, cybersecurity and privacy protection — Evaluation criteria for IT security”.
- [i.8] BSI CC-PP-0067 “Operating System Protection Profile”.

3 Definition of terms, symbols and abbreviations

Editor's Note: This Section Needs To Be Updated

3.1 Terms

This clause provides terms and definitions based on CEN/CLC JTC13 WG09's [i.6] work on terms and definitions, terms and definitions provided by ETSI EN 303 645/TS 103 701 [i.3] and by CEN/CLC EN 18031 [i.5] series, and informed by terms used in the Common Criteria [i.7] and the NIAP Operating System Protection Plan [i.8]guide.

For the purposes of the present document, the following terms apply.

Operating System (OS) is defined in European Commission Implementing Act 2025/2392.

Editor's Note: *The definition of Operating System is copied below for reference.* > Software products with digital elements that provide an abstract interface of the underlying hardware and control the execution of software, and that may provide services such as computing resource management and configuration, scheduling, input-output control, managing data, and providing an interface through which applications interact with system resources and peripherals. > This category includes but is not limited to real-time operating systems, general-purpose and special-purpose operating systems.

General Purpose Operating System: class of operating system designed to support a wide variety of workloads consisting of concurrent applications or services

NOTE: Typical characteristics of this category include support for third-party applications, support for multiple users, and security separation between users and their respective resources. General Purpose Operating Systems lack the operational constraints which define Special Purpose Operating Systems and Real Time Operating System (RTOS) that are typically used in routers, switches, and embedded devices.

Application Programming Interface (API): specification of routines, data structures, object classes, and variables that allows an application to make use of services provided by another software component, such as a library

NOTE: APIs are often provided for a set of libraries included with the platform.

system call interface: specification for the API between the application layer and the kernel or system layer

Input/Output (I/O): process or function for passing data to or from a given process over a specific interface

NOTE: Such I/O interfaces include, but are not limited to, serial ports, network ports, long-term storage devices including hard drives and flash drives, as well as human-interface ports such as display and audio devices.

Common Criteria (CC): Common Criteria for Information Technology Security Evaluation (International Standard ISO/IEC 15408) [i.7]

administrator: entity that is responsible for management activities, including setting policies that are applied by the enterprise on the operating system

NOTE: This administrator could be acting remotely through a management server, from which the system receives configuration policies. An administrator can enforce settings on the system which cannot be overridden by non-administrator users.

user: entity that is subject to configuration policies applied to the operating system by administrators

NOTE: On some systems under certain configurations, a normal user can temporarily elevate privileges to that of an administrator. At that time, such a user should be considered an administrator.

user account: identity created in an operating system with associated access controls and privileges

NOTE: Users may have multiple user accounts and user accounts may have multiple users.

threat actor: entity that can adversely affect the system through malicious or unauthorized activities

application: software that runs on a platform and performs tasks on behalf of the user or owner of the platform, as well as its supporting documentation

credential: data that establishes the identity of a user, e.g. a cryptographic key or password

Personally Identifiable Information (PII): any information about an individual maintained by an agency, including, but not limited to, education, financial transactions, medical history, and criminal or employment history and information which can be used to distinguish or trace an individual's identity, such as their name, government-issued identity numbers, date and place of birth, biometric records, etc., including any other personal information which is linked or linkable to an individual

sensitive data: sensitive data may include all user or enterprise data or may be specific application data such as PII, emails, messaging, documents, calendar items, contacts, credentials, and keys

data execution prevention: anti-exploitation feature of modern operating systems executing on modern computer hardware, which enforces a non-execute permission on pages of memory that are not code

NOTE: This prevents pages of memory from containing both data and instructions, which makes it more difficult for an attacker to introduce and execute code.

non-writable executable memory: anti-exploitation feature of modern operating systems executing on modern computer hardware, which enforces a non-write permission on pages of memory that are code

NOTE: This prevents modifying the instructions of running programs, which makes it more difficult for an attacker to introduce and execute code.

credential: data that establishes the identity of a user, e.g. a cryptographic key or password

Address Space Layout Randomization (ASLR): anti-exploitation feature which loads memory mappings into unpredictable locations

NOTE: ASLR makes it more difficult for an attacker to redirect control to code that they have introduced into the address space of a process.

Common Weakness Enumeration (CWE): community-developed list of software and hardware weaknesses that can become vulnerabilities <https://cwe.mitre.org/>

elevated privilege: level of access that allows accessing or changing security-relevant configuration, data, or functions on a system

command shell: text-based interface allowing execution of system programs

process isolation: techniques to prevent processes from accessing or changing each other's state

attack surface: user interfaces, target protocol interfaces and reachable data paths that can be attacked from inside or outside the system

principle of least privilege: design principle requiring that users, processes, and interfaces are granted only the minimum level of permission necessary to perform their legitimate functions, and nothing more

3.2 Symbols

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

Abbreviation	Description
OS	Operating system
MMU	Memory Management unit
I/O	Input/Output

For the purposes of the present document's risk analysis, the following abbreviations apply:

Abbreviation	Description
UC	Use Case
SP	Security Profile
RF	Risk Factor
SA	Security Assurance Level

4 Product context

Editor's Note: This Section Is Stable

4.1 Intended purpose and reasonably foreseeable use

The intended purpose of this product is to provide an abstract interface of the underlying hardware and control the execution of software, and that may provide services such as computing resource management and configuration, scheduling, input-output control, managing data, and providing an interface through which applications interact with system resources and peripherals.

Its reasonably foreseeable use is to facilitate the use of computing hardware and serve as a platform for other software.

4.2 Essential functions

The essential functions of the product may include functions such as: process isolation, memory isolation, I/O abstraction, device driver management, user authentication and access control, event logging, software management, device firmware management, secure updates, and more.

4.3 Product architecture

4.3.1 Overview

An operating system abstracts the hardware, allocates resources, and provides services to itself and any other software on the system. It often serves as a central organizing authority that controls access to system resources by various pieces of software, dividing up available resources among the applications and its own subsystems to meet implicit or explicit goals or constraints. An operating system often uses a large part of the resources of the system it runs on, but in return it simplifies the development and deployment of the overall system.

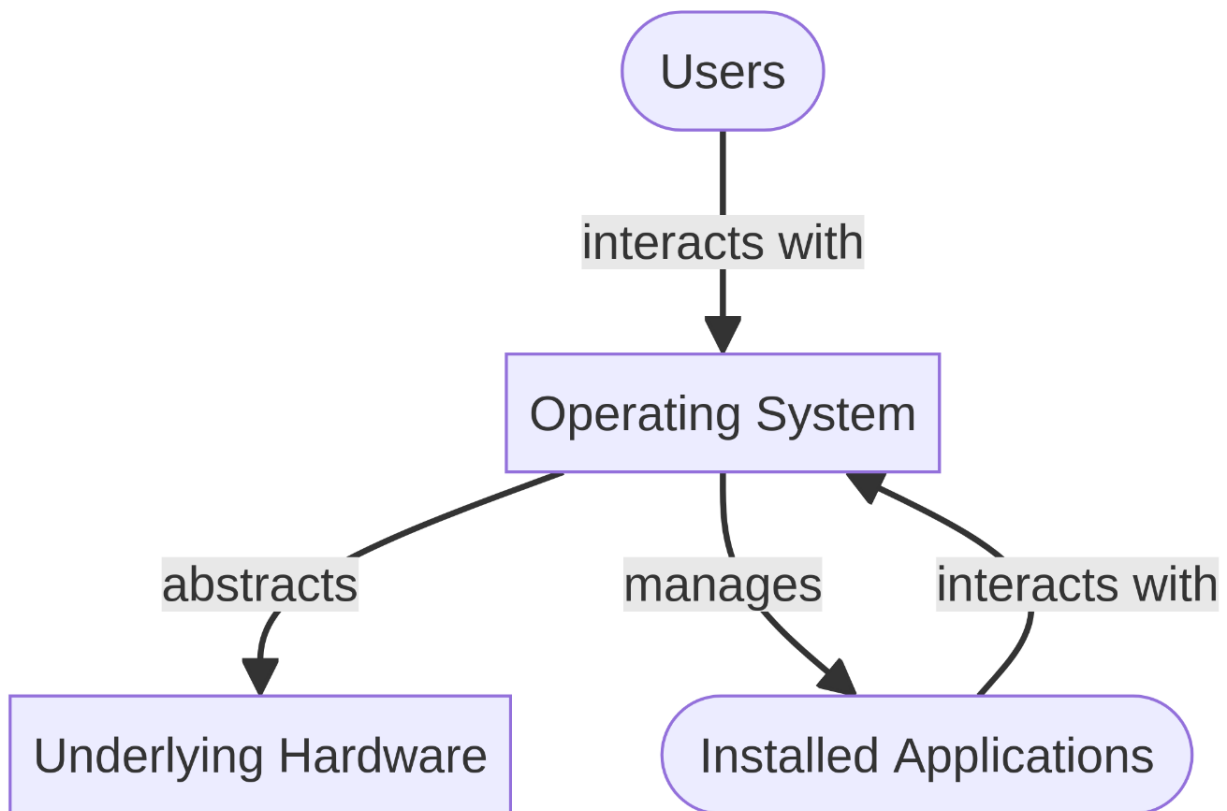


Figure 1: architecture diagram

Operating system architecture varies widely and depends on many factors, including the intended use case, the underlying platform, and the design philosophy of the developers. This overview will focus on the elements of operating system architecture that have a significant impact on the security functions and risk mitigations of an operating system.

4.3.2 Operating system security functions

The security of an operating system relies heavily on properly controlling the instructions executed by the system processor. Instruction flow is dependent on the program instructions, any data it accesses, and any hardware that has the ability to change either data or which instructions are being executed.

Example #1: a network interface adapter can write directly to system memory, potentially altering data values and thereby influencing the behavior of running programs or even causing unintended code execution.

Example #2: a USB controller can raise an interrupt which can cause the interrupt controller to force the CPU to switch to executing interrupt handling code, altering the instruction flow.

A fundamental building block of most operating systems is the principle of “privilege”. Privilege determine which system resources or functions a program is allowed to access. The operating system grants a specific set of privileges to itself, its subsystems, and user-level programs, ensuring that each component operates within defined boundaries to maintain system security and stability.

Generally, privileges are enforced using hardware features such as a memory management unit and processor-defined privilege levels. If the hardware does not provide these features, the operating system may use a best effort approach, such as relying on the compiler to generate code that is less likely to accidentally interfere with the functions of the operating system or other parts of the system. A best-effort approach is only acceptable in low-risk, low-impact use cases where all elements and users of the systems can be trusted to not deliberately attempt to compromise the system.

4.3.3 High-level operating system architectures

Operating systems architecture can vary in many ways which significantly affect security outcomes for the product. For this reason, the present document does not attempt to define specifics of operating system architecture. Some examples of differences include:

- the proportion of operating system code executed with different privileges
- whether application code is executed with the same privileges as the operating system
- the method of communication between processes with different privileges
- the degree of reliance on hardware capabilities to enforce privilege separation and process isolation

Each of these design variations makes different tradeoffs between security, performance, and ease of implementation.

A few common operating system architectures are described below for reference, but should not be taken normatively.

Monolithic kernel: The operating system kernel is one executable running at the highest processor privilege level in one memory domain, providing the majority of system-wide services. Applications are in separate memory domains and have a lesser privilege level. Applications request services and resources via system calls. Often a monolithic kernel supports modules, which allows parts of the kernel of the kernel to be added or removed at run-time, but they are usually sharing the same memory domain and privilege level.

Microkernel: The operating system kernel running at the highest processor privilege level provides a minimal set of resource allocation services, while many of the operating systems services are provided by separate executables with lower privileges.

Hybrid kernel: A mix of microkernel and monolithic kernel, where some operating systems services are provided in the central kernel and some are provided by applications. Subsystems that do not need to be high performance and are a frequent source of vulnerabilities are often moved into applications, such as printer drivers or file systems with complex features and/or low performance requirements.

Exokernel: The operating system does not abstract the resources of the system, it only manages resource allocation between different applications.

Unikernel: The operating system and the application are effectively a single executable. Often this is described as a library operating system: a set of library routines that an application can include and effectively become the operating system.

4.3.4 Access control mechanisms

Operating systems may control access to resources in different ways, including but not limited to:

Access control lists: Each resource has a list of users or processes that are allowed to access it.

Role-based access control: Users are assigned one or more roles, and roles have permissions associated with them.

Capabilities: Access to a resource is linked to a token which can be passed between processes.

4.3.5 Resource management

Operating systems may control resource management in different ways, including but not limited to imposing per-process limitations on processor time, memory allocation, storage usage, number of file descriptors, or number of process table entries.

Operating systems may implement limits on the number or proportion of specific resources that an application or thread may use, and may group these limits by user, process, process group, or other mechanism.

4.3.6 Scheduling

Operating systems may provide voluntary or involuntary switching between different processes, and may rely on hardware capabilities to improve or limit parallel process execution and isolation.

Some common models include:

Cooperative scheduling: Each thread runs until it voluntarily yields control of the processor to another thread. No thread is interrupted unless it explicitly yields the CPU.

Preemptive scheduling: Threads can be involuntarily suspended by the scheduler and replaced with other threads, and they may also voluntarily yield the CPU.

Operating system may perform scheduling based on many factors, such as but not limited to:

- Time spent executing during a previous time period
- Time since the thread was first marked runnable
- Explicit priorities associated with each thread
- Type of thread (kernel or application)
- Resource limits
- Performance considerations

4.4 Operational Environment

The operational environment of an operating system is highly varied. In general it includes at least a platform (virtual or physical), and may also include applications, separately shipped device drivers, device firmware, peripherals, and many other components. Operating systems may be a standalone software product on a single processor, or it may be part of suite of software products running on multiple processors. Many systems have multiple operating systems all managing different parts of the system at the same time.

4.5 Distribution of security functions

4.5.1 General

For each security requirement, a product may:

1. Provide all necessary security functions itself
2. Require security functions be provided by some other part of its context
3. Provide security functions for the use of other components

Most individual hardware components do not have a built-in method of secure firmware update, and rely on the presence of an operating system which can update the component's firmware securely.

4.5.2 Security functions provided outside the product

Operating systems often rely on essential external functionality to implement their necessary security functionality.

For example, many operating systems rely on hardware secure elements and a compatible boot manager prior to the beginning of the operating system's own execution in order for the operating system to execute securely.

4.5.3 Security functions provided to other components

Operating systems often provides essential security functions to other components of the system.

4.6 Users

Users of products may interact directly with the operating system, or the operating system may be integrated into a product in such a way that the end user does not interact directly with any functionality of the operating system.

4.7 Use Cases

Editor’s Note: The following list of use cases is an illustrative set of possible use cases selected to demonstrate the mechanics of this standard and provide clear guidance for the most common product categories.

Editor’s Note: Considering that Operating Systems provide an essential functionality for all Digital Products, it is not feasible to list in detail either all extant or all potential use cases for operating systems.

Editor’s Note: We anticipate that future revisions of this document may include additional use cases, such as for the following product scenarios: embedded devices, baseband management controllers, network interface cards, graphics cards, real-time applications, and special purpose operating systems.

4.7.1 UC-LR: Operating system for learning and research

- is not used for any purpose beyond learning and research
- does not store any sensitive or useful data
- security is provided entirely by the environment
- is highly modified by the user

4.7.2 UC-IoT-1: Non-internet-connected device such as a bluetooth speaker

- does not store any user-specific data
- has no means to connect directly to a public network
- not intended to support hardware, software, or operating system changes

4.7.3 UC-IoT-2: Internet-enabled power switch

- connects to a central service, operated by the device manufacturer, for remote data processing
- stores account information to authenticate to WiFi and to cloud service provider
- has a minimalistic interface, such as a single button for pairing and a reset button
- does not have accessible I/O ports

4.7.4 UC-IoT-3: Internet-connected “smart home” device

- e.g. a thermostat, fridge, or alarm system
- connects to a central service, operated by the device manufacturer, for remote data process
- stores account information to authenticate to WiFi and to cloud service provider
- does not support arbitrary file storage or end-user operating system configuration changes
- does not have accessible I/O ports
- may display personalized information, such as location-specific weather forecast
- serviced by trained professionals who do not modify software or hardware outside of manufacturer specifications

4.7.5 UC-RO-1: Consumer-grade home wireless router

- stores account information for authentication with ISP

- not intended for end-user hardware or software modification
- is exposed to the open internet

4.7.6 UC-OT-1: Business-grade remote door locking system

- does not store any user data
- not intended for hardware or software modification
- is not exposed to the open internet, and is only connected to trusted networks
- only serviced by professionals
- does not have accessible I/O ports
- hardware likely contains tamper-evident signals which operating system can rely on

4.7.7 UC-MOB-1: Personal mobile device

- stores highly sensitive personal information
- large number of sensors allow mass collection of sensitive personal data
- size and cost make it a common target of theft
- device usage is not limited to trusted locations and loss is foreseeable
- hardware and operating system configuration not intended for modification by users
- end-users frequently install software of uncertain provenance
- device frequently connects to untrusted networks
- device frequently collects user's location at all times
- device is often always on and always connected

4.7.8 UC-WE-1: Wearable health tracker

- e.g. a smart watch or step tracker
- stores information about a single user only
- stored information may be highly sensitive, and is likely to be strictly structured (not arbitrary files)
- does not have accessible I/O ports and is not user-modifiable
- connects to a central service, operated by the device manufacturer, for remote data processing
- connections are proxied by a trusted device, such as a mobile phone
- is not exposed to a public network

4.7.9 UC-PC-1: Personal computer in a fixed and generally safe location

- hardware, software and operating system may be configured and modified by the end-user
- the user may not be either highly skilled or an authorized representative of the manufacturer
- foreseeably connects to a public network and to low-trust local networks, but is not reachable from the open internet

- stores personal information and arbitrary files

4.7.10 UC-PC-2: Enterprise workstation in a fixed and generally safe location

- installed in an access-controlled workspace
- serviced by trained professionals who may modify both software and hardware
- connected to a public network with external mitigations, such as enterprise-grade firewalls
- connects to trusted local networks
- hardware likely contains tamper-evident indicators and secure elements for cryptographic storage
- used for web browsing
- stores business data, personal information and arbitrary files

4.7.11 UC-LA-1: Personal laptop

- hardware, software and operating system may be configured and modified by the end-user
- device is a foreseeable target of theft and tampering by untrusted 3rd parties
- stores personal information and arbitrary files
- unrestricted connection to a public network
- is frequently connected to untrusted networks
- hardware likely contains tamper-evident indicators and secure elements for cryptographic storage

4.7.12 UC-LA-2: Enterprise laptop

- hardware, software and operating system may be configured and modified by the end-user
- serviced by trained professionals who may modify both software and hardware
- device is a foreseeable target of theft and tampering by untrusted 3rd parties
- stores business data, personal information and arbitrary files
- unrestricted connection to a public network
- is frequently connected to untrusted networks
- hardware likely contains tamper-evident indicators and secure elements for cryptographic storage

4.7.13 UC-PS-1: Personal server

- one or a small number of trusted users
- installed in a fixed location at home or in a cohosting facility
- connected to a public network with a firewall
- connects to trusted local network
- limited access permitted from a public network for specific services
- semi-professional semi-automated management by one or a few people
- always stationary, access to hardware interfaces unlikely

4.7.14 UC-SE-1: Enterprise server in a datacenter with no user accounts

- installed in a monitored and secured facility
- serviced by trained professionals who may modify both software and hardware
- connected to a public network with external mitigations, such as enterprise-grade firewalls
- connects to trusted local networks
- hardware likely contains tamper-evident indicators and secure elements for cryptographic storage

4.7.15 UC-SE-2: Enterprise server in a datacenter with only trusted user accounts

- Same as UC-SE-2 but with trusted users

4.7.16 UC-SE-3: Enterprise server in a datacenter hosting many untrusted user accounts

- Same as UC-SE-2 but with untrusted users

5 Requirements specifications

5.1 Notes on the structure of the Requirements

Editor's Note: The CRA requires the manufacturer to keep all the documentation necessary to show that the tests were conducted. In Article 13 Rec. 22, MSA's are granted the right to request "all the information and documentation, in paper or electronic form, necessary to demonstrate the conformity of the product with digital elements and of the processes put in place by the manufacturer with the essential cybersecurity requirements set out in Annex I." The objective of these requirements is to provide manufacturers with sufficient guidance to consistently satisfy such requests from the market authorities.

5.1.1 Necessity of Requirements

Not all requirements are necessary for all products. The mapping table at the end of each requirement enumerates the set of risk factors and product use cases for which the requirements are necessary. See Annex C for more information.

5.1.2 Types of Technical Requirements

Testable Requirements: The essential quality of technical requirements is their capability to be verified by testing an implementation of the product. This verification ensures that the requirement can be assessed in an objective manner. If such verification is not practicable, the requirement is instead classified as a documentation requirement.

Documentation Requirements: For documentation requirements, the manufacturer shall provide supporting documentation, including but not limited to configuration files and documented organizational policies to ensure that compliance with the standard can be demonstrated even when direct product testing is not feasible.

NOTE: Simple process requirements (e.g., a statement that something has been done, without supporting evidence) are not acceptable and should be converted into technical requirements whenever possible and documentation requirements otherwise.

5.1.3 Assumptions Regarding Requirements

5.1.3.1 Testability

Manufacturers are already required to provide the ability to enable testing and collect output on the product as placed on the market, and will supply instructions for enabling and collecting test data.

5.1.3.2 Source Code

The market authorities may request source code access as part of a verification process, if necessary.

5.1.3.3 Mitigations

Mitigations are the technical means by which a technical requirement is satisfied. Mitigations should be tailored to the use case and take into account the foreseeable and expected skill level of the product's users, appropriate for the expected operational environment, and proportional to the foreseeable risk.

5.1.3.4 Risk Transferral

Some risks may be transferred partially or fully to other components of the final product, or to the user of the product. When that is the case, mitigations that transfer the risk will be included as an option to fulfill a technical requirement, depending on the use case and risk factors.

5.2 Technical security requirements specifications

5.2.1 General

This section is a list of technical requirements necessary to satisfy the CRA essential requirements. Each technical requirement can be satisfied by one or more potential mitigations. Each mitigation may or may not be appropriate for an individual use case. The following section will define which mitigations will be required, depending on risk factors and/or a use case.

NOT ALL MITIGATIONS ARE NECESSARY FOR ALL USE CASES. See Section 5.3 for the mappings of security profiles to mitigations and Annex C for additional information.

5.2.2 TR-NKEV: No known exploitable vulnerabilities at first use

5.2.2.1 Requirement

Recognizing that there may be vulnerabilities discovered between the time that a product is placed on the market and the time of that product's first use, and that the product should be free from known vulnerabilities both when first made available and when first used by a consumer, the product shall be able to be updated at the time of first use to address all known exploited vulnerabilities which were discovered after the product's placement on the market and before that first use.

5.2.2.2 MI-KEVD: Documentation for secure update before or during first use

The product shall be accompanied by documentation describing how the product may be securely updated, including how to update the product prior to, or as part of, first use.

- Applicability: Product expected use is long enough to require updates
- Reference: TR-NKEV
- Objective: Prevent exploitation of known exploited vulnerabilities at first use
- Preparation: Examine public or private vulnerability information sources and select a recently fixed vulnerability (preferably the most recently fixed)
- Activities: On a new product, carry out the initial secure update, scan the product to see if a recently fixed vulnerability has been fixed on the product, and examine the documentation for the required info
- Verdict: The secure update completes successfully, the most recently fixed vulnerability is fixed, and the documentation includes all the required information => PASS, otherwise FAIL
- Evidence: Documentation of vulnerability handling, documentation of how to securely update the product, the report for the selected vulnerability, description of how to scan for the vulnerability, log of vulnerability scan results

5.2.2.3 MI-KEVA: Automatic secure update before or during first use

The product shall implement automatic secure update by default before or during first use.

- Applicability: Product expected use is long enough to require updates
- Reference: TR-NKEV
- Objective: Prevent exploitation of known exploited vulnerabilities at first use
- Preparation: Examine public or private vulnerability information sources and select a recently fixed vulnerability (preferably the most recently fixed)

- Activities: Follow the instructions to install and use the product for the first time, scan the product to see if a recently fixed vulnerability has been fixed on the product, and examine the documentation for the required info
- Verdict: The secure update completes successfully, the most recently fixed vulnerability is fixed, and the documentation includes all the required information => PASS, otherwise FAIL
- Evidence: Documentation of vulnerability handling, documentation of how to securely update the product, the report for the selected vulnerability, description of how to scan for the vulnerability, log of vulnerability scan results

5.2.2.4 MI-KEVM: Documentation of mitigation of known exploitable vulnerabilities

The product's development and release process shall include a process to document known exploitable vulnerabilities in the product and their fixes or mitigations. The documentation for this process shall be compliant with the process described in [3] prEN 40000-1-3: "Cybersecurity requirements for products with digital elements – Vulnerability Handling". The product shall be compliant with this requirement if it:

1. has no known exploitable vulnerabilities
 2. has known exploitable vulnerabilities whose age is consistent with the specification of how long vulnerabilities may go unfixed after public disclosure, as described in the vulnerability handling procedure for the product
 3. for each detected vulnerability, has documentation of how the risk has been mitigated
- Reference: TR-NKEV
 - Objective: Prevent exploitation of known exploited vulnerabilities at first use
 - Preparation: Compile a list of known exploitable vulnerabilities in the product and its components
 - Activities: Compare the generated list of known exploitable vulnerabilities with the documentation of the known exploitable vulnerabilities that have been fixed or mitigated in the product
 - Verdict: No vulnerabilities found, or all reported vulnerabilities satisfy either the age or documentation requirement => PASS, otherwise FAIL
 - Evidence: Documented vulnerability handling policy, list of vulnerabilities, documentation of mitigations or age of vulnerability, correlation of list of vulnerabilities with documentation of mitigations or age of vulnerability

5.2.2.5 MI-KEVT: Testing for known exploitable vulnerabilities

The product shall be tested for all known exploitable vulnerabilities to demonstrate that each has been mitigated. The product shall be compliant with this requirement if it:

1. has no known exploitable vulnerabilities
 2. has known exploitable vulnerabilities whose age is consistent with the specification of how long vulnerabilities may go unfixed after public disclosure, as described in the vulnerability handling procedure for the product
 3. for each tested vulnerability, the test result shows that the vulnerability has been mitigated
- Reference: TR-NKEV
 - Objective: Prevent exploitation of known exploited vulnerabilities at first use
 - Preparation: Compile a list of known exploitable vulnerabilities in the product and its components, compile a list of known exploitable vulnerabilities that will be tested, collect tests for each one
 - Activities: On a new product, carry out a secure update, run the tests, and compare the results with the generated list of known exploitable vulnerabilities

- Verdict: No vulnerabilities found, or all reported vulnerabilities satisfy either the age or mitigation requirement => PASS, otherwise FAIL
- Evidence: Documented vulnerability handling policy, list of vulnerabilities, test results for each vulnerability or documentation of age of vulnerability, correlation of list of vulnerabilities with test results or documentation of age of vulnerability

5.2.2.6 MI-SCAN: No easily scannable known exploitable vulnerabilities

If automatable and freely-usable vulnerability scanners are available for the product, then the product shall satisfy the following with respect to the three (or fewer, if fewer than three are available) most comprehensive of such scanners:

1. has no vulnerabilities discovered by scans
 2. has discoverable exploitable vulnerabilities whose age is consistent with the specification of how long vulnerabilities may go unfixed after public disclosure, as described in the vulnerability handling procedure for the product
 3. for each detected vulnerability, has publicly available documentation explaining how the risk has been mitigated
- Reference: TR-NKEV
 - Objective: Prevent exploitation of known vulnerabilities at first use
 - Preparation: Select a set of tools meeting the requirements
 - Activities: On a new product, carry out a secure update, run the tools on the product, and examine the documentation for any reported vulnerabilities
 - Verdict: No vulnerabilities found, or all reported vulnerabilities satisfy either the age or documentation requirement => PASS, otherwise FAIL
 - Evidence: Documented vulnerability handling policy, list of vulnerability scanners selected, reports from each scanner, correlation of reports of discovered vulnerabilities with documentation of mitigations

5.2.3 TR-SSDD: Secure design and development

5.2.3.1 Requirement

The product shall be designed and developed in a secure manner.

5.2.3.2 MI-SSCA: Static source code analysis for memory errors

All security-relevant parts of the product shall be checked for memory errors using a source code analysis tool that detects code that may produce common memory errors, such as:

- buffer overflow
- out-of-bounds
- use after free
- double free
- use of uninitialized variables
- dereference of invalid pointer

The sufficiency of the source code analysis tool and the selected manner of running it shall be documented.

All warnings, annotations, or other method of suppressing warnings from the analysis tool shall be documented with a rationale for why it does not constitute an unacceptable risk.

- Reference: TR-SSDD
- Objective: Prevent unauthorized memory access
- Preparation: None
- Activities: Review the documentation on why the source code analysis tool is sufficient, how it is run, the source code for the product, the output of the source code analysis tool, and the documentation for any warnings or suppression of warnings
- Verdict: Sufficiency documentation is acceptable, the method of running the tool is consistent with rationale, the output of source code analysis tool is consistent with the source code, all warnings or suppression of warnings have convincing documentation for why they are an acceptable risk => PASS, otherwise FAIL
- Evidence: The documentation on why the source code analysis tool is sufficient, how it is run, the source code for the product, the output of the source code analysis tool, and the documentation for any warnings or suppression of warnings

5.2.3.3 MI-FZ95: Runtime code coverage checking with memory access error detection

The product shall be checked for memory errors by running a tool that exercises the functions of the product in an environment that permits measuring code coverage and detecting memory access errors. All memory errors detected shall be documented with a rationale for why it does not constitute an unacceptable risk.

- Reference: TR-SSDD
- Objective: Prevent unauthorized memory access
- Preparation: None
- Activities: Run the tool while measuring code coverage and monitoring for memory access errors until 95% code coverage has been reached
- Verdict: Code coverage was at least 95%, all reported memory errors are documented and justified => PASS, otherwise FAIL
- Evidence: Logs of code coverage tool, memory error report, documentation of any memory errors

5.2.3.4 MI-IMSL: Implement in a memory-safe language

The product's firmware and/or software shall be implemented in a memory-safe language. Any use of unsafe memory features shall be documented to explain why they are necessary and do not present a security risk.

- Reference: TR-SSDD, TR-MSAF
- Objective: Prevent unauthorized memory access
- Preparation: None
- Activities: Review source code to determine its language and what exceptions to memory safety exist
- Verdict: Source code is in a memory-safe language and the documentation of all uses of unsafe memory features convincingly demonstrates that each one of them does not present a security risk => PASS, otherwise FAIL
- Evidence: Source code, documentation of unsafe memory features

5.2.3.5 MI-BTIN: Boundary testing of inputs that may cause memory errors

The input fields of the product that may produce memory errors in the firmware or device driver shall be identified. The product shall be boundary tested for all such inputs while monitoring for memory errors. All memory errors detected shall be documented with a rationale for why it does not constitute an unacceptable risk.

- Reference: TR-SSDD, TR-MSAF
- Objective: Prevent unauthorized memory access
- Preparation: Identify input fields in the product that may produce memory errors
- Activities: Run a tool that tests input values that test the boundaries of the input values (minimum valid, maximum valid, minimum possible, maximum possible, off-by-one, etc.) while monitoring for memory errors
- Verdict: All boundary values tested and all memory errors detected are documented and justified => PASS, otherwise FAIL
- Evidence: Logs of boundary testing tool, memory error report, documentation of any memory errors

5.2.3.6 MI-SCFS: Secure compilation flags

All security-relevant firmware and software shall be compiled with secure compilation flags and options appropriate to the target platform and language. All compilation flags used shall be documented as to their rationale, along with any exceptions or limitations. Any exceptions to the flags or warnings shall be documented as to why they do not create an unacceptable risk.

- Applicability: Product implemented in a compiled language
- Reference: TR-SDDV
- Objective: Secure design and development
- Preparation: Document which flags should be used
- Activities: Review compilation flags, warnings, and documentation for exceptions
- Verdict: Documentation of flags exists, all warnings and exceptions are documented
- Evidence: Documentation of flags, build system files, documentation of warnings and exceptions

5.2.4 TR-MISO: Prevent local unauthorized access of memory-addressable security-relevant data

5.2.4.1 Requirement

The product shall protect memory addresses from unauthorized access by executables under the product's control, including the product itself. This includes system memory, storage addressable via memory mapping, memory for I/O devices, and anything else accessible via the memory-related instructions in the platform.

The product does not need to protect against unauthorized access by elements of the platform it is running on (e.g. CPU microcode, devices on the system bus, other operating systems in the device, a hypervisor). Future iterations of the standard may add this requirement for appropriate use cases.

5.2.4.2 MI-MMAC: Memory access control

The product shall implement mandatory hardware-enforced access control to memory to prevent unauthorized access of memory.

- Applicability: Has user accounts
- Reference: TR-MISO
- Objective: Prevent unauthorized memory access
- Preparation: List the methods of accessing memory and the types of access control to memory

- Activities: For each method of accessing memory and each type of access control to memory, attempt to use the method of accessing memory to gain access to memory that the executable is not authorized to access due to the access control
- Verdict: All memory accesses fail => PASS, otherwise FAIL
- Evidence: List of methods of accessing memory and types of access control, output of tests

5.2.4.3 MI-CCON: Prevent creation of more than one user account

The product shall prevent the creation of a user account if one already exists.

- Applicability: Has user accounts
- Reference: TR-MISO
- Objective: Prevent unauthorized access of memory
- Preparation: List all user accounts and verify there is exactly one
- Activities: Attempt to create a second user account, then list user accounts again
- Verdict: Creation of second user account fails and list of user accounts shows one account and is identical before and after test => PASS, otherwise FAIL
- Evidence: List of user accounts before and after test, output of test

5.2.4.4 MI-UCON: Prevent concurrent user account usage

The product shall prevent a user account from logging in if another user account is already logged in.

- Applicability: Has user accounts
- Reference: TR-MISO
- Objective: Prevent unauthorized access of memory
- Preparation: Create two accounts, log in to one account, list all logged in user accounts and verify there is exactly one, list all methods of logging in to a user account
- Activities: For each method of logging in to a user account, attempt to login as a second user account, then list logged in user accounts again
- Verdict: Login of second user account fails or is not possible, and list of user accounts logged in shows one account and is identical before and after test => PASS, otherwise FAIL
- Evidence: List of logged in user accounts before and after test, output of each test or evidence proving that logging in as a second user is impossible

5.2.4.5 MI-PMSC: Prevent memory leaks through microarchitectural side channels in provided executables

The product shall implement mechanisms to prevent the executables it provides from leaking memory data to unauthorized users through known exploitable microarchitectural side channels (MASCs), such as via the observing the time of cache access for various operations including:

- speculative execution/loads/stores
- branch prediction
- out-of-order execution
- shared multithreading resources

- address translation
- memory access patterns
- prefetching

Test:

- Reference: TR-MISO
- Objective: Prevent unauthorized reads of memory
- Preparation: List known MASC leaks on supported platform
- Activities: For each type of MASC leak, run a test using the best known techniques to exploit the MASC on a system-provided executable
- Verdict: All tests fail to extract data that they do not have authorization to read => PASS, otherwise FAIL
- Evidence: Output of each test

5.2.4.6 MI-TRMD Transfer risk of microarchitectural side channel data leaks to user

The documentation provided to the user shall document the risk of microarchitectural side channel data leaks and give appropriate guidance to the user on how they may mitigate the risk.

- Applicability: (for requirements that depend on a feature)
- Reference: TR-MISO
- Objective: Prevent unauthorized reads of memory
- Preparation: None
- Activities: Read documentation provided with the product
- Verdict: Documentation sufficiently describes the risks and mitigations => PASS, otherwise FAIL
- Evidence: Documentation provided with the product

5.2.5 TR-MSAF: Mitigate memory safety errors

5.2.5.1 Requirement

The product shall appropriately mitigate risks due to memory safety errors.

5.2.6.2 Default Preparation, Verdict, and Evidence

Most memory safety mitigations have the same Verdict and Evidence:

- Preparation: None
- Verdict: each involved thread fails to read or write the target data and takes a segmentation fault, has error handling code executed, or is terminated in all tests => PASS, otherwise FAIL
- Evidence: error messages, log message, or the product reboots or halts

For each mitigation grouped under requirement TR-MSAF, for each field Preparation, Verdict, or Evidence, if it is not specified for that test, then the above Preparation, Verdict, or Evidence field shall apply.

5.2.5.3 MI-MSAF-1: Stack exhaustion detection

Both kernel and userspace threads shall reject writes beyond the end of the stack.

- Reference: TR-MSAF
- Objective: Prevent thread from writing beyond end of stack
- Activities: For each of kernel and userspace, write beyond the end of the stack

Guidance: Two methods of exhausting stack memory include allocating a very large object on the stack, and performing an unbounded recursive function call.

5.2.5.4 MI-MSAF-2: Stack linear buffer overflow detection

Both kernel and userspace threads shall reject stack buffer writes that go beyond the end of the stack frame.

- Reference: TR-MSAF
- Objective: Prevent thread from writing beyond end of stack
- Activities: For each of kernel and userspace, write beyond the end of the stack frame

5.2.5.5 MI-MSAF-3: Array bounds checking

Both kernel and userspace threads shall reject writes to fixed-size arrays that are beyond the end of the array.

- Reference: TR-MSAF
- Objective: Prevent thread from writing beyond the end of a fixed-size array
- Activities: For each of kernel and userspace, write beyond the end of a fixed-size array

5.2.5.6 MI-MSAF-4: Heap linear buffer overflow detection

Both kernel and userspace threads shall reject writes beyond the bounds of allocated heap memory.

- Reference: TR-MSAF
- Objective: Prevent thread from writing beyond the end of heap memory
- Activities: For each of kernel and userspace, for each type of heap memory, allocate a fixed size from each class of heap memory, write beyond it

5.2.5.7 MI-MSAF-5: Heap use-after-free access prevention

Both kernel and userspace threads shall reject use of allocated memory that has been freed.

- Reference: TR-MSAF
- Objective: Prevent thread from using memory that was allocated then freed
- Activities: For each of kernel and userspace, allocate from heap memory, free it, then try to read it, repeat but with a write

5.2.5.8 MI-MSAF-6: Heap free checking

Both kernel and userspace threads shall reject freeing of memory that was allocated and previously freed.

- Reference: TR-MSAF
- Objective: Prevent thread from freeing memory that is already free
- Activities: For each of kernel and userspace, allocate from heap memory, free it, then free again

5.2.6 TR-LMII: Limit incident impact

5.2.6.1 Requirement

The product shall implement appropriate mitigations to limit incident impact.

5.2.6.2 Default Preparation, Verdict, and Evidence

Most memory safety mitigations have the same Verdict and Evidence:

- Preparation: None
- Verdict: each involved thread fails to read or write the target data and takes a segmentation fault, has error handling code executed, or is terminated in all tests => PASS, otherwise FAIL
- Evidence: error messages, log message, or the product reboots or halts

For each mitigation grouped under requirement TR-LMII, for each field Preparation, Verdict, or Evidence, if it is not specified for that test, then the above Preparation, Verdict, or Evidence field shall apply.

5.2.6.3 MI-MZRO-1: Stack memory zeroing

Both kernel and userspace threads shall zero-initialize all stack memory before use.

- Reference: TR-LMII
- Objective: Prevent attacker from exploiting erroneous use of uninitialized stack memory
- Activities: For each of kernel and userspace, sequentially call 2 functions that allocate the same amount of memory, fill the first with non-zero values and return, and during second function call, read the stack contents back
- Verdict: stack contents are all zero on second call
- Evidence: contents of stack before the first function return, contents of stack during the second function call

5.2.6.4 MI-MZRO-2: Heap memory zeroing

Both kernel and userspace threads shall zero-initialize all heap memory before use.

- Reference: TR-LMII
- Objective: Prevent attacker from exploiting erroneous use of uninitialized heap memory
- Activities: For each of kernel and userspace, allocate heap memory, fill with a non-zero value, free it, allocate it again in a deterministic way to get the same heap region, and read back the contents
- Verdict: memory contents are all zero on second call
- Evidence: contents of allocated memory before the free, contents of allocated memory after second allocation

5.2.6.5 MI-MRWX-1: Prevent writes to executable and read-only data memory

Both kernel and userspace threads shall reject writes to executable and read-only data memory

- Reference: TR-LMII
- Objective: Prevent writes to executable and read-only data memory
- Activities: For each of kernel and userspace, for each portion of executable and non-writable data regions, write to it

5.2.6.6 MI-MRWX-2: Prevent execution of non-kernel code memory

Kernel threads shall prevent execution of non-kernel code memory.

- Reference: TR-LMII
- Objective: Mitigate exploits that use execution of arbitrary memory
- Activities: For each class of non-code memory in the kernel (e.g. stack, heap, read-only data), copy a trivial return-only function into the memory, and attempt to execute each one

5.2.6.7 MI-ASLR Address space layout randomization

The product shall enable Address Space Layout Randomization (ASLR) by default for all executables, including the kernel, if any.

- Applicability: Platform has an MMU and product implements virtual memory
- Reference: TR-LMII
- Objective: Exploit mitigation
- Preparation: None
- Activities: For every executable, examine the object file to determine if ALSR is enabled. For one non-kernel executable (if any) and one kernel executable (if any), run the executable twice and read the base addresses of the text, stack, heap, and shared libraries where applicable.
- Verdict: All executables have ALSR enabled, base addresses collected for executables differ between runs => PASS, else FAIL
- Evidence: Output of scan for ALSR enabled, base addresses collected

5.2.6.8 MI-MRCO: Mitigate reference counter overflow

Both kernel and userspace threads shall mitigate the effects of reference counter overflows

- Reference: TR-LMII
- Objective: Prevent exploitation of bugs in reference counting to overflow the counter to zero, causing a free and subsequent use-after-free accesses
- Activities: For each of kernel and userspace, set resource reference counter to 1 less than maximum representable value, increment it twice
- Verdict: reference counter does not overflow and resource is permanently pinned (no longer can be freed)
- Evidence: test output showing reference counter values before and after the operation, allocation status of the resources

5.2.6.9 MI-NKAM: Prevent unintentional kernel access to userspace memory

Kernel threads shall prevent cross-privilege memory access.

- Applicability: Product has multiple privilege levels
- Reference: TR-LMII
- Objective: Mitigate exploits that use kernel privileges to access arbitrary userspace memory
- Activities: For each of read, write, and execute operations, use a kernel thread to attempt to use the operation on memory regions that are mapped to a different privilege level without going through dedicated memory access routines

Guidance: The most common privilege levels are kernel and userspace.

5.2.6.10 MI-PLLC: Prevent linked list corruption

Both kernel and userspace threads shall check the consistency of the previous and next pointers it manipulates when adding or deleting an item to or from a linked list and reject the operation if they are not consistent.

- Reference: TR-LMII
- Objective: Prevent linked list corruption
- Activities: For each of kernel and userspace, add or delete an item to an uninitialized list

5.2.6.11 MI-CFIN: Control flow integrity

Both kernel and userspace threads shall protect saved function and return pointers from overwrite

- Reference: TR-LMII
- Objective: Mitigate exploits by preventing overwrite of function and return pointers
- Activities: For each of kernel and userspace, save a function pointer to the heap, overwrite it with a different function, make indirect call to the saved function pointer, then repeat but with a return address that was stored to the stack

Guidance: This mitigation can be implemented via software (e.g. ASan) or hardware (e.g. Pointer Authentication), or validating transitions of expected control flow graph (e.g. KCFI, Shadow Stack).

5.2.6.12 MI-MPMT: Memory protection using memory tagging

Both kernel and userspace threads shall use hardware-supported memory tagging to reject erroneous memory accesses.

- Reference: TR-LMII
- Objective: Mitigate exploits by preventing memory errors
- Activities: For each of kernel and userspace, allocate 2 adjacent memory regions with separate tags. Attempt to read and write memory with a positive offset into trailing region from leading region's tagged pointer. Attempt to read and write with negative offset into leading region using trailing region's tagged pointer. Free a region and read and write to the region using the original tagged pointer.

5.2.7 TR-MINI: Minimize impact on other devices and services

5.2.7.1 Requirement

The product shall implement appropriate mitigations to minimize impact on other devices and services.

Editor's Note: We hope that there will be additional contributions to this section in the future.

5.2.7.2 MI-MDOC: Document transfer of risk of minimizing impact to operating environment

The product shall be accompanied by documentation informing the user of the transfer of risk for minimizing impact on other devices and services.

- Reference: TR-MINI
- Objective: Minimize impact on other devices and services
- Activities: Examine the documentation
- Verdict: Transfer of risk documented in a manner appropriate to the user => PASS, otherwise FAIL

- Evidence: Documentation, analysis of documentation

5.2.7.3 MI-MNET: Minimize negative impact of network transmission

The product shall minimise its negative impact on other products or services via the data it transmits on the network. Each source of network data shall be documented, along with the ways it can interfere with other products or services, and methods the product uses to minimise that interference.

- Reference: TR-MINI
- Objective: Minimise negative impact on others
- Preparation: List all sources of transmitted network data on the product
- Activities: For each method of sending network data, examine the documentation of the ways it can interfere with other products or services, and what methods the product uses to minimise that interference
- Verdict: Every method of sending network data is documented with ways it can interface and methods used to minimise => PASS, otherwise FAIL
- Evidence: All configuration files for network services, documentation of network services and their impact and methods to minimise it, internal lists of listening ports, results of an external port scan

5.2.7.4 MI-MAMP: Minimize negative impact of network traffic amplification

The product shall mitigate abuse of network services that amplify network traffic in manner that can be used to attack other devices. Each network service and its associated mitigations shall be documented.

- Reference: TR-MINI
- Objective: Minimise negative impact on others
- Preparation: List all network services that return responses larger than the recieved packet without authorization of the source
- Activities: For each network service, examine the documentation of the steps taken to limit access, rate-limit, or otherwise mitigate the use of the service in traffic amplication attacks
- Verdict: Every method of sending network data is documented with how its impact on others has been mitigated => PASS, otherwise FAIL
- Evidence: All configuration files for network services, documentation of network services and their impact and methods to minimise it, internal lists of listening ports, results of an external port scan, calculation of traffic amplification factors

5.2.8 TR-SDEF: Secure by default configuration

5.2.8.1 Requirement

The product shall operate in a secure configuration by default.

5.2.8.2 MI-ADEF: Authorization required by default to access security-relevant assets

The product's default state shall require appropriate authorization to access all security-relevant assets. Appropriate authorization depends on the use case and the asset. For example, an autogenerated device-specific cryptographic key should not be readable without appropriate authorization.

- Reference: TR-SDEF
- Objective: Find any unauthorized access to security relevant assets in default configuration
- Preparation: List all interfaces allowing access to security-relevant assets

- Activities: For each interface, attempt to access security-relevant assets without appropriate authorization and record whether access was allowed or not
- Verdict: If every interface does not allow access without appropriate authorization => PASS, otherwise => FAIL
- Evidence: List of interfaces allowing access to security-relevant assets, record of activities used to attempt unauthorized access to security-relevant assets, log of results of attempts
- Applicability: if the product's intended purpose is for integration into another product, this mitigation may be implemented by the integrator

5.2.8.3 MI-PDDI-1: Document how to protect access to debug and management interfaces

All debug and management interfaces on the product shall be documented, and the documentation shall specify means to protect or disable them.

- Applicability: This mitigation is for products intended for integration into subsequent products.
- Reference: TR-SDEF
- Objective: Secure by default
- Preparation: Examine the documentation for how to protect or disable the debug/management interfaces of the product
- Activities: Examine the product for undocumented debug/management interfaces, then follow the instructions in the documentation to disable or protect each documented interface, then attempt to access the interface without authorization
- Verdict: All debug/management interfaces are documented as to how to disable or protect them, and no interfaces are accessible without authorization after following the documentation to protect or disable them => PASS, otherwise => FAIL
- Evidence: Pictures of the product, list of discovered interfaces, comparison with documentation, notes as to which are documented how to disable/protect, logs of protect/disable actions, logs of attempts to access interfaces after protected or disabled

5.2.8.4 MI-PDDI-2: Protect or disable physical access to debug and management interfaces

All debug and management interfaces which can be accessed by an agent with physical access to the device the product is installed on shall be protected or disabled by default, unless necessary for backward compatibility. Documentation regarding the removal of such protections by an appropriately sophisticated user may be provided, and shall include information regarding the risks.

- Reference: TR-SDEF
- Objective: Secure by default
- Preparation: Examine the documentation of the network- and localhost-accessible interfaces of the product and follow the instructions to mitigate the risk of any necessary unprotected or enabled interfaces
- Activities: As an unprivileged process running on the system, attempt to access the system's local debug and management interfaces and make unauthorized changes. Additionally, scan accessible memory and inter-process-communication mechanisms for undocumented debug and management interfaces.
- Verdict: No undocumented interfaces are found and no interfaces can be accessed without authorization other than those documented as necessary and the instructions to the user are sufficient => PASS, otherwise => FAIL
- Evidence: List of interfaces, log of attempts to access

5.2.8.4 MI-PDDI-3: Protect or disable local software access to debug and management interfaces

All debug and management interfaces which can be accessed by processes running on the system shall be protected or disabled by default, unless necessary for backward compatibility. Documentation regarding the removal of such protections by an appropriately sophisticated user may be provided, and shall include information regarding the risks.

- Reference: TR-SDEF
- Objective: Secure by default
- Preparation: Examine the documentation of the network- and localhost-accessible interfaces of the product and follow the instructions to mitigate the risk of any necessary unprotected or enabled interfaces
- Activities: As an unprivileged process running on the system, attempt to access the system's local debug and management interfaces and make unauthorized changes. Additionally, scan accessible memory and inter-process-communication mechanisms for undocumented debug and management interfaces.
- Verdict: No undocumented interfaces are found and no interfaces can be accessed without authorization other than those documented as necessary and the instructions to the user are sufficient => PASS, otherwise => FAIL
- Evidence: List of interfaces, log of attempts to access

5.2.8.5 MI-PDDI-4: Protect or disable network access to debug or management interfaces

All debug and management interfaces accessible via the network shall be protected or disabled by default, unless necessary for backward compatibility. Documentation regarding the removal of such protections by an appropriately sophisticated user may be provided, and shall include information regarding the risks.

- Reference: TR-SDEF
- Objective: Secure by default
- Preparation: Examine the documentation of the network accessible interfaces of the product and follow the instructions to mitigate the risk of any necessary unprotected or enabled interfaces
- Activities: Using a network scanner, scan the product for both documented and undocumented debug or remote management interfaces and determine whether they are enabled or protected
- Verdict: No undocumented interfaces are found and no interfaces can be accessed without authorization other than those documented as necessary and the instructions to the user are sufficient => PASS, otherwise => FAIL
- Evidence: List of interfaces, log of attempts to access

5.2.9 TR-SCUD: Secure updates

5.2.9.1 Requirement

The product shall be securely updateable by the user.

5.2.9.2 MI-SCHL: Low security updates provided by operational environment

The technical documentation provided with the product shall document that the operational environment shall provide a method of receiving notifications of secure updates from the manufacturer, retrieving the updates, verifying the updates, and applying them to the product. The secure update method shall satisfy the "Low" security level for the product supplying it.

- Reference: TR-SCUD
- Objective: Secure updates

- Activities: Assess the documentation provided with the product
- Verdict: Documentation describes requirements for the secure updates provided by the operational environment => PASS, otherwise FAIL
- Evidence: Documentation and analysis of completeness

5.2.9.3 MI-SCHM: Medium security updates provided by operational environment

The technical documentation provided with the product shall document that the operational environment shall provide a method of receiving notifications of secure updates from the manufacturer, retrieving the updates, verifying the updates, and applying them to the product. The secure update method shall satisfy the “Medium” security level for the product supplying it.

- Reference: TR-SCUD
- Objective: Secure updates
- Activities: Assess the documentation provided with the product
- Verdict: Documentation describes requirements for the secure updates provided by the operational environment => PASS, otherwise FAIL
- Evidence: Documentation and analysis of completeness

5.2.9.4 MI-SCHH: High security updates provided by operational environment

The technical documentation provided with the product shall document that the operational environment shall provide a method of receiving notifications of secure updates from the manufacturer, retrieving the updates, verifying the updates, and applying them to the product. The secure update method shall satisfy the “High” security level for the product supplying it.

- Reference: TR-SCUD
- Objective: Secure updates
- Activities: Assess the documentation provided with the product
- Verdict: Documentation describes requirements for the secure updates provided by the operational environment => PASS, otherwise FAIL
- Evidence: Documentation and analysis of completeness

5.2.9.5 TODO

Editor’s Note: We are waiting on an anticipated submission of additions to this section from an ETSI member who has unexpectedly become unavailable.

5.2.7 TR-AUTH: Authentication and access control

Editor’s Note: We anticipate that future revisions of this document will include state-of-the-art authentication requirements. This section should reference authentication and access control standards.

5.2.7 TR-CDST: Confidentiality of data stored on the product

5.2.7.1 Requirement

The product shall protect data stored on the product from unauthorized access.

5.2.7.2 MI-CDST: Protect confidentiality of data stored on the product

Editor's Note: We have included only a high-level mitigation, and anticipate that more detailed and specific mitigations will be added later.

The product shall protect data stored on the product from unauthorized access.

- Reference: TR-CDST
- Objective: Confidentiality of data
- Preparation: List all types of data that may be stored on the product that should not be readable without authorization, what methods of ensuring confidentiality are appropriate for each type, all methods of accessing that data available to an attacker based on the risk assessment, and what the allowable authorization methods are for that access method
- Activities: For each type of data and each access mechanism, determine the method of ensuring confidentiality used, and attempt to read the data without authorization
- Verdict: If all methods of ensuring confidentiality match the type of the data stored, and all the attempts to read confidential data without authorization fail => PASS, otherwise => FAIL
- Evidence: Logs of determination of type of data and method of confidentiality and attempts to read confidential data without authorization

Guidance: Data may be protected by the environment, permissions, encryption, salting and hashing, offline storage, or hardware-backed secrets.

5.2.8 TR-CDTX: Confidentiality of data transmitted by product

5.2.8.1 Requirement

The product shall protect data transmitted by the product from unauthorized access.

5.2.8.2 MI-CDTX: Protect confidentiality of data transmitted by product

Editor's Note: We have included only a high-level mitigation, and anticipate that more detailed and specific mitigations will be added later.

The product shall protect data transmitted by the product from unauthorized access.

- Reference: TR-CDTX
- Objective: Confidentiality of data
- Preparation: List all types of data that may be transmitted on the product that should not be readable without authorization, what methods of ensuring confidentiality are appropriate for each type, all methods of accessing that data available to an attacker based on the risk assessment, and what the allowable authorization methods are for that access method
- Activities: For each type of data and each access mechanism, determine the method of ensuring confidentiality used, and attempt to read the data without authorization
- Verdict: If all methods of ensuring confidentiality match the type of the data transmitted, and all the attempts to read confidential data without authorization fail => PASS, otherwise => FAIL
- Evidence: Logs of determination of type of data and method of confidentiality and attempts to read confidential data without authorization

Guidance: Data transmitted may be protected by the environment or encryption.

5.2.8.3 MI-DOCC: Document transfer of risk of confidentiality of data transmitted by product

The product shall be accompanied by documentation informing the user of the transfer of risk for protecting the confidentiality of data transmitted by the product.

- Reference: TR-CDTX
- Objective: Protect data confidentiality
- Activities: Examine the documentation
- Verdict: Transfer of risk documented in a manner appropriate to the user => PASS, otherwise FAIL
- Evidence: Documentation, analysis of documentation

5.2.9 TR-CRYP: Encryption

Editor's Note: We anticipate that future revisions of this document will include state-of-the-art encryption requirements, including references to appropriate encryption standards not already included in the Agreed Cryptographic Mechanism and CRA Addendum.

5.2.10 TR-IDST: Integrity of data stored on the product

5.2.10.1 Requirement

The product shall protect the integrity of data stored on the product from unauthorized modification and report corruption.

Guidance: Integrity may be protected by the environment, permissions, duplication, backups, and/or checksums.

5.2.10.2 MI-IDST: Protect integrity of data stored on the product

Editor's Note: We have included only a high-level mitigation, and anticipate that more detailed and specific mitigations will be added later.

The product shall protect the integrity of data stored on the product from unauthorized modification.

- Reference: TR-IDST
- Objective: Integrity of data
- Preparation: List all types of data that may be stored on the product that should not be modifiable without authorization, what methods of protecting integrity are appropriate for each type, all methods of modifying that data available to an attacker based on the risk assessment, and what the allowable authorization methods are for that modification method
- Activities: For each type of data and each access mechanism, determine the method of protecting integrity used, and attempt to modify the data without authorization
- Verdict: If all methods of ensuring integrity match the type of the data stored, and all the attempts to modify protected data without authorization fail => PASS, otherwise => FAIL
- Evidence: Logs of determination of type of data and method of integrity and attempts to modify protected data without authorization

5.2.10.3 MI-DCST: Detect corruption of data stored

Editor's Note: We have included only a high-level mitigation, and anticipate that more detailed and specific mitigations will be added later.

The product shall detect corruption of the data stored on the product.

- Reference: TR-IDST
- Objective: Integrity of data
- Preparation: List all types of data that may be stored on the product whose corruption should be detected and what methods of detecting corruption are appropriate for each type
- Activities: For each type of data and method of detecting corruption, corrupt the data in a way that the method will detect
- Verdict: If all methods of detecting corruption match the type of the data stored, and all the corruptions of data are detected => PASS, otherwise => FAIL
- Evidence: Logs of determination of type of data and corruptions of data

5.2.11 TR-IDTX: Integrity of data transmitted by the product

5.2.11.1 Requirement

The product shall detect corruption of the data transmitted by the product.

Guidance: Integrity may be protected by the environment, permissions, duplication, backups, and/or checksums.

5.2.11.2 MI-DCTX: Detect corruption of data transmitted by the product

Editor's Note: We have included only a high-level mitigation, and anticipate that more detailed and specific mitigations will be added later.

The product shall detect corruption of the data transmitted by the product.

- Reference: TR-IDTX
- Objective: Integrity of data
- Preparation: List all types of data that may be transmitted by the product whose corruption should be detected and what methods of detecting corruption are appropriate for each type
- Activities: For each type of data and method of detecting corruption, corrupt the data in a way that the method will detect
- Verdict: If all methods of detecting corruption match the type of the data transmitted, and all the corruptions of data are detected => PASS, otherwise => FAIL
- Evidence: Logs of determination of type of data and corruptions of data

5.2.12 TR-DMIN: Data Minimization

5.2.12.1 Requirement

The product shall minimize the data processed.

5.2.12.2 MI-DJST: Document and justify processed data

All sources of data processed by the product in its secure-by-default configuration shall be documented. All sources of data processed shall have a documented rationale for why its processing is necessary for the functioning of the product in its secure-by-default configuration.

- Reference: TR-DMIN
- Objective: Minimize data processed

- Preparation: List all potential sources of data for the product. For each source of data, identify a method to detect whether the product is processing data from that source.
- Activities: Using the list of sources of data, and the method to detect whether the product is processing data from that source, list all sources of data processed. Compare to the documented list.
- Verdict: All sources of processed data are documented, including rationale => PASS, otherwise => FAIL
- Evidence: List of sources of data, documentation of each source of data, list of sources of data processed, connection between each discovered source of processed data to its documentation

5.2.13 TR-AVAI: Availability

5.2.13.1 Requirement

The product shall protect the availability of essential and core functions.

5.2.13.2 MI-AVNT: Availability of network services

The product shall protect the availability of essential and core network services through mitigation of denial-of-service attacks.

- Reference: TR-AVAI
- Objective: Protect availability of network functions
- Preparation: List all network services and identify essential and core network services
- Activities: For each essential or core network service, examine the documentation for how the product sufficiently mitigates denial-of-service attacks for its risk assessment
- Verdict: Every essential or core network service is documented and the mitigations are sufficient => PASS, otherwise FAIL
- Evidence: All configuration files for network services, documentation of network services and the ways to mitigate a denial-of-service attack on it, internal lists of listening ports, results of an external port scan

5.2.13.3 MI-WDOG: Watchdog and self-initiated reset

The product shall implement a mechanism to trigger an automatic reset when it detects that it is no longer able to perform its functions.

- Reference: TR-AVAI
- Objective: Availability
- Preparation: Document the conditions that indicate the product cannot perform its functions
- Activities: Cause each of the conditions to occur and observe whether the product resets
- Verdict: Every condition triggers an automatic reset => PASS, otherwise FAIL
- Evidence: Documentation, log messages

5.2.13.4 MI-FDRP: Fast packet drop

TODO: Write mitigation requiring the product to do validity checks on packets from both the network and the user in order of cheapest to most expensive so it can drop invalid packets with as little resource usage as possible.

5.2.13.5 MI-LMEM: Limit memory usage

TODO: Write mitigation requiring the product limit memory usage triggered by user input via network or local access.

5.2.13.6 MI-FAIR: Fair resource usage and prioritization

TODO: Write mitigation requiring the product implement some form of ensuring fair resource usage by multiple sources of input, including the ability to prioritize some sources of input.

5.2.13.7 MI-DOST: Document risk transfer to operational environment for denial of service

TODO: Write mitigation documenting that the operational environment must provide denial of service protection, such as an external or internal firewall, fair queueing or filtering, a proxy, etc.

5.2.14 TR-LMAS: Minimize exposed interfaces

5.2.14.1 Requirement

The manufacturer shall minimize exposed interfaces in the default configuration of the product in all operating modes, including initial configuration, during initialization, while in use, while shutting down or paused, or after reset.

5.2.14.2 MI-JSTY: Document and justify exposed interfaces

All exposed interfaces on the product in any state that is part of its reasonably foreseeable use or misuse in its secure-by-default configuration shall be documented. Every interface shall have a documented rationale for why its exposure is necessary for the functioning of the product in its secure-by-default configuration.

- Reference: TR-LMAS
- Objective: Limit attack surface
- Preparation: List all types of interfaces on the product that may be exposed to an attacker, whether enabled or disabled. For each type of interface, identify a method to list all exposed interfaces of that type. List all states of the product with different exposed interfaces of the product in its secure-by-default configuration, including but not limited to initial configuration, startup, in use, idle, shutdown, and reset, if applicable. For each distinct exposed interface in each state, describe the interface and why it has to be enabled by default.
- Activities: Using the list of types of interfaces, the list of states of the product, and the method to list all exposed interfaces of that type, list all exposed interfaces in each state. Compare to the documented list.
- Verdict: All discovered interfaces are documented, including rationale => PASS, otherwise => FAIL
- Evidence: List of types of interfaces, list of product states, documentation of each exposed interface, output of methods to list all exposed interfaces, connection between each discovered interface to its documentation

5.2.15 TR-LOGG: Logging and monitoring

5.2.15.1 Requirement

The product shall record security-relevant internal events, including but not limited to changes to configuration and access or modification of data and functions. The product shall provide an opt-out mechanism.

5.2.15.2 MI-LOGG: Logging

The product shall record log messages indicating security-relevant internal events in an internal or external log. The log messages shall not include any confidential information such as PII, secrets, or credentials, or any information which might reasonably be expected to include such items.

- Reference: TR-LOGG
- Objective: Monitoring and recording security-relevant events
- Preparation: List all types of security-relevant internal events
- Activities: For each type of security-relevant internal event, trigger the event
- Verdict: For each triggered event, the log contains a message indicating the event, log message does not include any information likely to be confidential => PASS, otherwise FAIL
- Evidence: Method of triggering events, log messages with annotations

Guidance: One type of event whose log message must take care to not accidentally include a secret is failed password authentication attempts. Since people often type their password into the username field, including the username field in the log message may result in including a secret in the log message.

5.2.16 TR-SCDL: Secure deletion

5.2.16.1 Requirement

The product shall provide a method of deleting all user data and settings and resetting the product to its secure-by-default configuration.

Guidance: Overwriting all user-writable storage or encrypting all user data and deleting the key are two secure deletion mechanisms.

5.2.16.2 MI-RSET: Secure deletion via reset

The product shall reset to its secure-by-default state after a power cycle or reset command.

- Applicability: Product has the capability for the user to write data and/or settings
- Reference: TR-SCDL
- Objective: Secure deletion
- Preparation: Document every kind of stored data or setting that may be changed by the user on the product, how to store it on the product, and how to read it from the product
- Activities: For each kind of user data or setting that may be stored and changed by the user on the product, write an instance of the data or setting stored on the product that is different from the default and read it from the product; once all kinds of data have been written and read, power cycle or reset the product, and read each kind of data again
- Verdict: If any data or setting is the same for both of the reads => FAIL, otherwise => PASS
- Evidence: Record of each type of data or setting, what data or setting was written, what data or setting was returned by the first read, and what data or setting was returned by the second read, comparison of each one

5.2.16.3 MI-INST: Secure deletion via reinstallation

The product shall reset to its secure-by-default state after a reinstallation that securely deletes all previous user data or settings.

- Applicability: Product has the capability for the user to write data and/or settings

- Reference: TR-SCDL
- Objective: Secure deletion
- Preparation: Document every kind of data or setting that may be stored and changed by the user on the product, how to store it on the product, and how to read it from the product
- Activities: For each kind of user data or setting that may be stored and changed by the user on the product, write an instance of the data or setting stored on the product that is different from the default and read it from the product; once all kinds of data have been written and read, reinstall the product with the secure delete option, and read the data or settings again
- Verdict: If any data or setting is the same for both of the reads => FAIL, otherwise => PASS
- Evidence: Record of each type of data or setting, what data or setting was written, what data or setting was returned by the first read, and what data or setting was returned by the second read, comparison of each one

5.2.16.4 MI-DELE: Secure deletion via secure deletion function

The product shall reset to its secure-by-default state after the secure deletion function is used.

Editor's Note: this section should be clarified so that the method of deletion depends upon the sensitivity of data stored.

- Applicability: Product has the capability for the user to write data and/or settings
- Reference: TR-SCDL
- Objective: Secure deletion
- Preparation: Document every kind of data or setting that may be stored and changed by the user on the product, how to store it on the product, and how to read it from the product
- Activities: For each kind of user data or setting that may be stored and changed by the user on the product, write an instance of the data or setting stored on the product that is different from the default and read it from the product; once all kinds of data have been written and read, activate the secure deletion function, and read the data or settings again
- Verdict: If any data or setting is the same for both of the reads => FAIL, otherwise => PASS
- Evidence: Record of each type of data or setting, what data or setting was written, what data or setting was returned by the first read, and what data or setting was returned by the second read, comparison of each one

5.2.17 TR-SDTR: Secure data read and transfer

5.2.17.1 Requirement

The product shall provide a method to read all data and settings from the product, and if provided, securely transfer data and settings to another product.

5.2.17.2 MI-SDRF: Secure data read from product

The product shall provide a method by which an authorized user can securely read all data and settings from the product.

- Applicability: Product has the capability for the user to write data and/or settings
- Reference: TR-SDTR
- Objective: Secure data read
- Preparation: List all data and settings

- Activities: For each kind of data or setting, read the data or setting as an authorized user, then attempt read the data or setting as an unauthorized user, if any exists
- Verdict: All data and settings can be read by the authorized user, and no data or setting can be read by an unauthorized user => PASS, otherwise FAIL
- Evidence: List of data and settings, log message showing success or failure of each read by the authorized user and, if applicable, the unauthorized user

5.2.17.3 MI-SDTR: Secure data transfer to another product

If the product provides a method to transfer data and settings to another product, it shall do so securely.

- Applicability: Product has the capability for the user to write data and/or settings and to transfer them to another product.
- Reference: TR-SDTR
- Objective: Secure data transfer
- Preparation: Prepare methods by which an unauthorized user could read the data during transfer as outlined in the risk assessment
- Activities: Read the data or settings, initiate the data transfer, attempt to read or alter the transferred data and settings as an unauthorized user, read the new data and settings on the target product
- Verdict: No data or settings could be read or altered by an unauthorized user, and the data and settings read from the original product and target product are the same wherever technically possible => PASS, otherwise FAIL
- Evidence: List of data and settings, log messages from the attempts to read or alter data as the unauthorized user, data and settings as read from the source product and as read from the target product, comparison explaining technical reasons for any differences in the two versions

5.2.18 TR-VULH: Vulnerability handling

5.2.18.1 Requirement

The product shall have vulnerability handling processes compliant with [3] prEN 40000-1-3: “Cybersecurity requirements for products with digital elements – Vulnerability Handling”.

5.2.18.2 MI-VULH-1: Vulnerability Handling in the Product

The product shall have vulnerability handling processes compliant with [3] prEN 40000-1-3: “Cybersecurity requirements for products with digital elements – Vulnerability Handling”.

- Applicability: (for requirements that depend on a feature)
- Reference: TR-VULH
- Objective: Vulnerability handling
- Activities: Review documentation associated with vulnerability handling.
- Verdict: Vulnerability handling documentation is compliant with [3] prEN 40000-1-3: “Cybersecurity requirements for products with digital elements – Vulnerability Handling” => PASS, otherwise FAIL
- Evidence: Vulnerability handling documentation, comparison with [3] prEN 40000-1-3: “Cybersecurity requirements for products with digital elements – Vulnerability Handling”

5.2.18.3 MI-VULH-2: Enabling Vulnerability Handling in Integrated Products

When the product is intended for integration into subsequent products in a supply chain, system vulnerabilities may have a particularly high impact on the security characteristics of the final product. Therefore, manufacturers of operating systems intended for integration in subsequent products have a responsibility to enable the vulnerability handling processes of manufacturers which depend upon them. This is accomplished by sharing, as appropriate to the specific risks, details of the operating system's components to enable downstream manufacturers to participate in coordinated vulnerability handling procedures described in [3] prEN 40000-1-3: "Cybersecurity requirements for products with digital elements – Vulnerability Handling".

- Applicability: any operating system intended for integration in subsequent products, rather than use by an end-user
- Reference: TR-VULH
- Objective: Vulnerability handling
- Activities: Review product's SBOM for detailed lists of third-party components and verify the accuracy of identifiers of those components. If applicable and permissible, apply scanning and analysis techniques to the product to verify that the supplied SBOM is accurate and complete.
- Verdict: Product's SBOM contains accurate identifiers for third-party components which can be verified from appropriate sources and unlisted third-party components are not discovered through product inspection => PASS, otherwise FAIL
- Evidence: Logs from product analysis and comparison to supplied SBOM

5.3 Risk Mitigation Sets

5.3.1 General

Each risk mitigation is only necessary for the security profiles (see clause C.6.2) that require it to treat a risk. This clause lists all mitigations that are necessary for each security profile.

5.3.2 SP-LR required mitigations

None.

5.3.3 SP-IoT-1 required mitigations

None.

5.3.4 SP-IoT-2 required mitigations

1. SSCA
2. SCFS
3. MMAC
4. ADEF
5. LOGG
6. KEVA
7. KEVM
8. (KEVT or SCAN)
9. (SUAP or SUA0)

10. VULH
11. PDDI-1
12. AUTH
13. DOCC
14. DJST
15. DOST
16. (MDOC or MAMP)
17. SUDC
18. CDTX
19. CRYP
20. IDTX
21. DMIN

5.3.5 SP-IoT-3 required mitigations

1. SSCA
2. SCFS
3. MMAC
4. ADEF
5. LOGG
6. KEVA
7. KEVM
8. (KEVT or SCAN)
9. (SUAP or SUA0)
10. VULH
11. PDDI-1
12. AUTH
13. DOCC
14. DJST
15. DOST
16. LMEM
17. (MDOC or MAMP)
18. SUDC
19. CDTX
20. CRYP
21. IDTX

22. DMIN

5.3.6 SP-RO-1 required mitigations

1. SSCA
2. (FZ95 or BTIN or IMSL)
3. SCFS
4. MMAC
5. ASLR
6. MSAF-*
7. MZRO-*
8. MRWX-*
9. NKAM
10. PLLC
11. MRCO
12. ADEF
13. JSTY
14. LOGG
15. KEVA
16. KEVM
17. (KEVT or SCAN)
18. (SUAP or SUA0)
19. VULH
20. PDDI-1
21. PDDI-4
22. AUTH
23. CDTX
24. DCTX
25. DOCC
26. DJST
27. DOST
28. AUTH
29. AVNT
30. FDRP
31. LMEM
32. FAIR

33. MNET
34. MAMP
35. SUDC
36. CDTX
37. CRYP
38. IDTX
39. DMIN

5.3.7 SP-OT-1 required mitigations

1. SSCA
2. (FZ95 or BTIN or IMSL)
3. SCFS
4. MMAC
5. ASLR
6. MSAF-*
7. MZRO-*
8. MRWX-*
9. NKAM
10. PLLC
11. MRCO
12. ADEF
13. JSTY
14. LOGG
15. KEVA
16. KEVM
17. (KEVT or SCAN)
18. (SUAP or SUA0)
19. VULH
20. ADEF
21. PDDI-1
22. PDDI-2
23. PDDI-4
24. AUTH
25. DOCC
26. DJST

27. DOST
28. AVNT
29. FDRP
30. LMEM
31. FAIR
32. (MDOC or MAMP)
33. SUDC
34. CDTX
35. CRYP
36. IDTX
37. DMIN

5.3.8 SP-MOB-1 required mitigations

1. SSCA
2. (FZ95 or BTIN or IMSL)
3. SCFS
4. MMAC
5. ASLR
6. MSAF-*
7. MZRO-*
8. MRWX-*
9. NKAM
10. PLLC
11. MRCO
12. ADEF
13. JSTY
14. LOGG
15. KEVA
16. KEVM
17. (KEVT or SCAN)
18. (SUAP or SUA0)
19. VULH
20. ADEF
21. PDDI-2
22. PDDI-4

23. AUTH
24. CDTX
25. DCTX
26. DOCC
27. DJST
28. DOST
29. AVNT
30. FDRP
31. LMEM
32. FAIR
33. MNET
34. MAMP
35. SUDC
36. CDTX
37. CRYP
38. IDTX
39. DMIN

5.3.9 SP-WE-1 required mitigations

1. SSCA
2. SCFS
3. MMAC
4. ADEF
5. JSTY
6. LOGG
7. (KEVD or KEVA)
8. KEVM
9. (SUVF or SUAP or SUOE or SUAQ)
10. VULH
11. ADEF
12. PDDI-1
13. PDDI-2
14. PDDI-4
15. AUTH
16. DOCC

17. DJST
18. DOST
19. (MDOC or MAMP)
20. SUDC
21. CDTX
22. CRYP
23. IDTX
24. DMIN

5.3.10 SP-PC-1 required mitigations

1. SSCA
2. (FZ95 or BTIN or IMSL)
3. SCFS
4. MMAC
5. ASLR
6. MSAF-*
7. MZRO-*
8. MRWX-*
9. NKAM
10. PLLC
11. MRCO
12. ADEF
13. JSTY
14. LOGG
15. KEVA
16. KEVM
17. (KEVT or SCAN)
18. (SUAP or SUA0)
19. VULH
20. PDDI-1
21. PDDI-3
22. PDDI-4
23. AUTH
24. CDTX
25. DCTX

26. DOCC
27. DJST
28. DOST
29. LMEM
30. MNET
31. MAMP
32. SUDC
33. CDTX
34. CRYP
35. IDTX
36. DMIN

5.3.11 SP-PC-2 required mitigations

1. SSCA
2. (FZ95 or BTIN or IMSL)
3. SCFS
4. MMAC
5. ASLR
6. MSAF-*
7. MZRO-*
8. MRWX-*
9. NKAM
10. PLLC
11. MRCO
12. ADEF
13. JSTY
14. LOGG
15. KEVA
16. KEVM
17. (KEVT or SCAN)
18. (SUAP or SUA0)
19. VULH
20. PDDI-1
21. PDDI-3
22. PDDI-4

23. AUTH
24. CDTX
25. DCTX
26. DOCC
27. DJST
28. DOST
29. AVNT
30. FDRP
31. LMEM
32. FAIR
33. MNET
34. MAMP
35. SUDC
36. CDTX
37. CRYP
38. IDTX
39. DMIN

5.3.12 SP-LA-1 required mitigations

1. SSCA
2. (FZ95 or BTIN or IMSL)
3. SCFS
4. MMAC
5. ASLR
6. MSAF-*
7. MZRO-*
8. MRWX-*
9. NKAM
10. PLLC
11. MRCO
12. ADEF
13. JSTY
14. LOGG
15. KEVA
16. KEVM

17. (KEVT or SCAN)
18. (SUAP or SUA0)
19. VULH
20. ADEF
21. PDDI-2
22. PDDI-3
23. PDDI-4
24. AUTH
25. CDTX
26. DCTX
27. DOCC
28. DJST
29. DOST
30. LMEM
31. MNET
32. MAMP
33. SUDC
34. CDTX
35. CRYP
36. IDTX
37. DMIN

5.3.13 SP-LA-2 required mitigations

1. SSCA
2. (FZ95 or BTIN or IMSL)
3. SCFS
4. MMAC
5. ASLR
6. MSAF-*
7. MZRO-*
8. MRWX-*
9. NKAM
10. PLLC
11. MRCO
12. ADEF

13. JSTY
14. LOGG
15. KEVA
16. KEVM
17. (KEVT or SCAN)
18. (SUAP or SUA0)
19. VULH
20. ADEF
21. PDDI-2
22. PDDI-3
23. PDDI-4
24. AUTH
25. CDTX
26. DCTX
27. DOCC
28. DJST
29. DOST
30. AVNT
31. FDRP
32. LMEM
33. FAIR
34. MNET
35. MAMP
36. SUDC
37. CDTX
38. CRYP
39. IDTX
40. DMIN

5.3.14 SP-PS-1 required mitigations

1. SSCA
2. (FZ95 or BTIN or IMSL)
3. SCFS
4. MMAC
5. ASLR

6. MSAF-*
7. MZRO-*
8. MRWX-*
9. NKAM
10. PLLC
11. MRCO
12. ADEF
13. JSTY
14. LOGG
15. KEVA
16. KEVM
17. (KEVT or SCAN)
18. (SUAP or SUA0)
19. VULH
20. PDDI-1
21. PDDI-3
22. PDDI-4
23. AUTH
24. CDTX
25. DCTX
26. DOCC
27. DJST
28. DOST
29. LMEM
30. MNET
31. MAMP
32. SUDC
33. CDTX
34. CRYP
35. IDTX
36. (TRMD or PMSC)
37. DMIN

5.3.15 SP-SE-1 required mitigations

1. SSCA

2. (FZ95 or BTIN or IMSL)
3. SCFS
4. MMAC
5. ASLR
6. MSAF-*
7. MZRO-*
8. MRWX-*
9. NKAM
10. PLLC
11. MRCO
12. ADEF
13. JSTY
14. LOGG
15. KEVA
16. KEVM
17. (KEVT or SCAN)
18. (SUAP or SUA0)
19. VULH
20. PDDI-1
21. PDDI-3
22. PDDI-4
23. AUTH
24. CDTX
25. DCTX
26. DOCC
27. DJST
28. DOST
29. AVNT
30. FDRP
31. LMEM
32. FAIR
33. MNET
34. MAMP
35. SUDC
36. CDTX

37. CRYP
38. IDTX
39. DMIN

5.3.16 SP-SE-2 required mitigations

1. SSCA
2. (FZ95 or BTIN or IMSL)
3. SCFS
4. MMAC
5. ASLR
6. MSAF-*
7. MZRO-*
8. MRWX-*
9. NKAM
10. PLLC
11. MRCO
12. ADEF
13. JSTY
14. LOGG
15. KEVA
16. KEVM
17. (KEVT or SCAN)
18. (SUAP or SUA0)
19. VULH
20. PDDI-1
21. PDDI-3
22. PDDI-4
23. AUTH
24. CDTX
25. DCTX
26. DOCC
27. DJST
28. DOST
29. AVNT
30. FDRP

31. LMEM
32. FAIR
33. MNET
34. MAMP
35. SUDC
36. CDTX
37. CRYP
38. IDTX
39. (TRMD or PMSC)
40. DMIN

5.3.17 SP-SE-3 required mitigations

1. SSCA
2. (FZ95 or BTIN or IMSL)
3. SCFS
4. MMAC
5. ASLR
6. MSAF-*
7. MZRO-*
8. MRWX-*
9. NKAM
10. PLLC
11. MRCO
12. ADEF
13. JSTY
14. LOGG
15. KEVA
16. KEVM
17. (KEVT or SCAN)
18. (SUAP or SUA0)
19. VULH
20. PDDI-1
21. PDDI-3
22. PDDI-4
23. AUTH

24. CDTX
25. DCTX
26. DOCC
27. DJST
28. DOST
29. AVNT
30. FDRP
31. LMEM
32. FAIR
33. MNET
34. MAMP
35. SUDC
36. CDTX
37. CRYP
38. IDTX
39. (TRMD or PMSC)
40. DMIN

6 Conformity Assessment

Editor's Note: It has not yet been determined by consensus among rapporteurs whether assessment guidance should be placed in Clause 5 requirements or moved to Clause 6, here. So, for now, we have opted to leave the assessment guidance adjacent to the requirement they are assessing as this provides, in our view, improved readability. They may be moved here in the future.

Annex A (informative): Mapping between the present document and CRA requirements

Editor's Note: The following table, which maps technical security requirements from clause 5 of the present document to essential cybersecurity requirements in Annex I of the CRA, has the purpose of assisting with the identification of missing technical security requirements.

CRA requirement	Technical security requirements(s)
No known exploitable vulnerabilities	NKEV
Secure design, development, production	SSDD, LMII
Secure by default configuration	SDEF
Secure updates	SCUD
Authentication and access control mechanisms	AUTH*
Confidentiality protection	MISO, LMII, CDST, CDTX, CRYP*
Integrity protection for data and configuration	MISO, IDST, IDTX
Data minimization	DMIN
Availability protection	AVAI, LMII
Minimize impact on other devices or services	MINI, SDEF, AVAI, SSDD, LMII
Limit attack surface	MISO, LMAS, SSDD, LMII
Exploit mitigation by limiting incident impact	MISO, LMII, AVAI, SSDD
Logging and monitoring mechanisms	LOGG
Secure deletion and data transfer	SCDL, SDTR
Vulnerability handling	VULH

* *waiting on cross-vertical*

Annex B (informative): Relationship between the present document and any related ETSI standards (if any)

ETSI EN 303 645: “CYBER; Cyber Security for Consumer Internet of Things: Baseline Requirements”[i.3] provided some terms and definitions.

Annex C (informative):
Risk identification and assessment methodology

C.1 Assets

C.1.1 Data

- Logs
- Secrets/credentials
- Executables (kernel, apps, libraries, etc.)
- Configuration
- Firmware images
- User data

C.1.2 Product functions

- Resource allocation
 - Memory management
 - Scheduling
 - Storage quotas
 - Other resource usage limits
- Isolation
 - Memory protection
 - Storage protection
 - Other permissions
- Abstract I/O
 - Network stack
 - File systems
 - Video
 - Sound
 - Input devices (mouse, keyboard)
- Hardware communication
 - Device drivers
- Power management
- Hardware management
- Configuration
 - Software to run
 - Hardware configuration
 - I/O configuration (network, etc.)
- Initialization

- Initialize hardware
- Start software services
- Authentication
- Authorization
- Software management
 - Software verification
 - Software installation
 - Security updates
 - Software upgrade
 - Software removal
 - Firmware upgrades
 - Load kernel modules
- Logging
- Monitoring/notifications

C.2 Risk factors

C.2.1 General comments regarding risk factors

Risk factors determine which mitigation(s) satisfy each of the technical requirements in clause 5.2.

Manufacturers determine the level of each risk factor via the development of a threat model and risk profile based on the intended and foreseeable use and misuse of the operating system.

Risk factors may increase the likelihood of an incident, increase the impact of an incident, or both. As a result, different mitigation strategies may be more or less relevant to different risk factors.

The overall risk related to each use case should be considered, and is calculated by combining risk factors affecting both likelihood and impact of an incident.

C.2.2 RF-NUSR: Number of User Accounts

The number of user accounts of end-users expected on the system, excluding administrator accounts.

- NUSR-0: foreseeable use does not include user accounts for end-users
- NUSR-1: foreseeable use is only one user account for an end-user
- NUSR-2: foreseeable use of the operating system is multiple user accounts for end-users

C.2.3 RF-CUSR: User Account Concurrency

The number of user accounts expected to use the system concurrently, including administrator accounts if they are configurable or accessible by end-users.

- CUSR-0: foreseeable use is one authenticated end-user using the device at a time, including authentication by physical access
- CUSR-1: foreseeable use of the operating system is small number of authenticated users simultaneously active on the operating system who are trusted not to actively attempt to compromise the system
- CUSR-2: foreseeable use of the operating system is multiple authenticated untrusted users simultaneously active on the operating system

C.2.4 RF-PPII: Potential for Collection of Personally Identifiable Information

Potential for collection of personally identifiable information about an individual person.

- PPII-0: foreseeable use includes no or incidental collection of PII
- PPII-1: foreseeable use includes collection of moderate amounts of PII
- PPII-2: foreseeable use includes collection of extensive amounts of PII by default

C.2.5 RF-SNDS: Sensitivity of Data Stored

Sensitivity of data stored, as measured by impact of loss of its integrity, confidentiality, or availability.

- SNDS-0: foreseeable use includes no or incidental storage of sensitive data
- SNDS-1: foreseeable use includes storing moderate amounts of sensitive data
- SNDS-2: foreseeable use includes storing extensive amounts of sensitive data by default

C.2.6 RF-SNDT: Sensitivity of Data Transmitted

Sensitivity of data transmitted, as measured by impact of loss of its integrity, confidentiality, or availability.

- SNDT-0: foreseeable use includes no or incidental transmission of sensitive data
- SNDT-1: foreseeable use includes transmission of moderate amounts of sensitive data
- SNDT-2: foreseeable use includes transmission of extensive amounts of sensitive data by default

C.2.7 RF-SENF: Sensitivity of Functions

Sensitivity of functions of device, as measured by impact of loss of its integrity, confidentiality, or availability.

- SENF-0: foreseeable use includes no or incidental provision of sensitive functions
- SENF-1: foreseeable use may provide arbitrary sensitive functions
- SENF-2: foreseeable use provides sensitive functions by default

C.2.8 RF-PHYS: Physical Access by Threat Actors to the Device

Exposure of the device to physical access by users.

- PHYS-0: foreseeable use is only in environments without physical exposure to untrusted users
- PHYS-1: foreseeable use includes incidental physical exposure to untrusted users
- PHYS-2: foreseeable use includes regular physical exposure to untrusted users

C.2.9 RF-UEIN: Processing of Untrusted External Inputs

Exposure to untrusted external inputs that are processed by the platform.

- UEIN-0: only used in environments without processing of untrusted external inputs
- UEIN-1: may incidentally process untrusted external inputs
- UEIN-2: used primarily to process untrusted external inputs

C.2.10 RF-LOSS: Probability of Loss of the Device

Likelihood of loss or theft of the device, allowing threat actors unlimited physical access to the device.

- LOSS-0: foreseeable use is in a device with no or incidental loss likelihood
- LOSS-1: foreseeable use is in a device with moderate loss likelihood
- LOSS-2: foreseeable use is in a device with a high loss likelihood, such as devices which are common targets of theft such as mobile phones

C.2.11 RF-HWMD: Hardware Modifiability by End Users

Likelihood that the hardware of the platform will be changed from its secure-by-default state.

- HWMD-0: foreseeable use limited to devices with hardware that is not modifiable by end-users
- HWMD-1: foreseeable use includes hardware modifications by skilled administrators
- HWMD-2: foreseeable use includes hardware modification by unskilled users

C.2.12 RF-SWMD: Software Modifiability by End Users

Likelihood that the software on the platform (including firmware) will be changed from its secure-by-default state.

- SWMD-0: foreseeable use only allows the installation of trusted and verified software, such as updates
- SWMD-1: foreseeable use allows for the installation of arbitrary software or for substantial modification of pre-installed software
- SWMD-2: foreseeable use actively encourages and facilitates the installation of frequently malicious software

C.2.13 RF-DVCS: Untrusted Peripheral Devices

Likelihood of untrusted peripheral devices being attached to the platform via a connection that is a plausible attack vector, such as by USB or PCI bus.

- DVCS-0: foreseeable use has no accessible peripheral ports
- DVCS-1: foreseeable use includes only trusted and safe peripheral devices
- DVCS-2: foreseeable use allows for arbitrary peripheral device attachment

C.2.14 RF-TNET: Access to a Public Network

Likelihood that the device will initiate connections to public networks.

- TNET-0: foreseeable use has no mechanism to reasonably connect to a public network
- TNET-1: foreseeable use allows internet access for only highly restricted functions, such as retrieving security updates
- TNET-2: foreseeable use allows for arbitrary access to a public network, such as by browsing the web

C.2.15 RF-FNET: Accessed From Untrusted Networks Including a Public Network

Likelihood that the device will be exposed to incoming traffic from public networks.

- FNET-0: foreseeable use is limited to trusted and private networks
- FNET-1: foreseeable use includes untrusted local networks but not the open internet
- FNET-2: foreseeable use includes being connected directly to the open internet

C.2.16 RF-CONF: Configurability

Degree of security-relevant configuration change of the operating system necessary for use.

- CONF-0: foreseeable use does not require storing operating system configuration changes
- CONF-1: foreseeable use involves operating system configuration changes only by skilled administrators
- CONF-2: foreseeable use of the operating system includes configuration changes by end-users

C.2.17 RF-ADMN: Administration

Availability and skill of administrators.

- ADMN-0: foreseeable use does not require administration
- ADMN-1: foreseeable use always has skilled administrators available on call

- ADMN-2: foreseeable use may involve unskilled administrators

C.2.18 RF-SUPP: Support and Foreseeable Updates

How long the product is expected to be in use, and whether the product is expected to be updated throughout its life cycle.

- SUPP-0: foreseeable use does not require that the operating system be updated at any point in its lifecycle
- SUPP-1: foreseeable use includes the installation of updates by end-users with access to the operating system
- SUPP-2: foreseeable use necessitates that the manufacturer provide frequent, automatic, and/or time-sensitive updates to the product, and may reasonably include a requirement for over-the-air updates.

C.3 Assumptions

Assumptions can be updated to be less stringent as more use cases and mitigations are added to the standard.

C.3.1 AS-PP: Proper platform

The platform the product runs on is trustworthy. The OS may choose to detect and/or correct hardware errors.

C.3.2 AS-PA: Proper administrator

The product administrator is not intentionally hostile and is engaging in good faith efforts to administer the system properly.

C.3.3 AS-LP: Attacker has limited physical access to product

An attacker will have only temporary physical access to the product.

C.3.4 AS-LR: Attacker has limited resources

An attacker has the resources available to a small group of skilled individuals, without the backing of large corporations, nation-states, or immense wealth.

C.4 Threats and risk assessments of threats

C.4.1 General

The approach to listing threats is to separate them by mitigation so that they may be associated with mitigations more directly.

C.4.2 Risk assessment methodology

Risk factor levels for each security profile are determined by reading the descriptions for each risk factor level and choosing the one that most accurately represents the highest risk for the intended purpose and reasonably foreseeable use and misuse of the product, as specified by the manufacturer.

For each threat, a formula based on the risk factor levels is used to calculate the Likelihood and Impact of the threat, on a scale of Low, Medium, and High.

For each threat, both likelihood and impact must be Low before the risk is considered sufficiently mitigated. If the calculated levels are not already Low, then mitigations must be applied until they are both Low. The mitigation sets that will accomplish this are listed in each threat description.

C.4.3 TH-UEVU: Unknown exploitable vulnerabilities

Attacker may use unknown exploitable vulnerabilities in the product implementation to get unauthorized access to product assets.

Risk factors	Likelihood	Security profiles
$\max(\text{NUSR}, \text{CUSR}, \text{SENF}, \text{PHYS}, \text{FNET}) = 0$	Low	LR, IoT-1
all others	Medium	IoT-2, IoT-3, WE-1
$\max(\text{NUSR}, \text{CUSR}, \text{SENF}, \text{PHYS}, \text{FNET}) = 2$	High	RO-1, OT-1, MOB-1, PC-*, LA-*, PS-1, SE-*

Risk factors	Impact	Security profiles
$\max(\text{SNDS}, \text{SNDT}, \text{SENF}) = 0$	Low	LR, IoT-1
$\max(\text{SNDS}, \text{SNDT}, \text{SENF}) = 1$	Medium	IoT-2, IoT-3
$\max(\text{SNDS}, \text{SNDT}, \text{SENF}) = 2$	High	RO-1, OT-1, MOB-1, WE-1, PC-*, LA-*, PS-1, SE-*

Requirements that mitigate this threat: SSDD, SDEF, MSAF, LMII, LMAS, LOGG

Mitigations for Likelihood:

- Medium to Low: SSCA, SCFS, MMAC, ADEF
- High to Low: SSCA, (FZ95 or BTIN or IMSL), SCFS, MMAC, ASLR, MSAF-*, MZRO-*, MRWX-*, NKAM, PLLC, MRCO, ADEF, JSTY

Mitigations for Impact:

- Medium to Low: LOGG
- High to Low: LOGG

C.4.4 TH-KEVU: Known exploitable vulnerabilities

Attacker may use known exploitable vulnerabilities in the product implementation to get unauthorized access to product assets.

Risk factors	Likelihood	Security profiles
$\text{ADMN} = 0$ or $\text{SUPP} = 0$	Low	LR, IoT-1
all others	Medium	WE-1
$\text{ADMN} = 2$ & $\text{SUPP} = 2$	High	IoT-2, IoT-3, RO-1, OT-1, MOB-1, PC-*, LA-*, PS-1, SE-*

Risk factors	Impact	Security profiles
$\max(\text{SNDS}, \text{SNDT}, \text{SENF}) = 0$	Low	LR, IoT-1
$\max(\text{SNDS}, \text{SNDT}, \text{SENF}) = 1$	Medium	IoT-2, IoT-3
$\max(\text{SNDS}, \text{SNDT}, \text{SENF}) = 2$	High	RO-1, OT-1, MOB-1, WE-1, PC-*, LA-*, PS-1, SE-*

Requirements that mitigate this threat: NKEV, SSDD, SDEF, MSAF, LMII, LMAS, LOGG

All mitigations from TH-UEVU apply (using that requirement's risk formula), in addition to:

Mitigations for Likelihood:

- Medium to Low: (KEVD or KEVA), KEVM, (SUVP or SUAP or SUOE or SUA0), VULH
- High to Low: KEVA, KEVM, (KEVT or SCAN), (SUAP or SUA0), VULH

C.4.5 TH-UAPP: Unauthorized access to product assets via unprotected physical interfaces in default configuration

Attacker may use unprotected debug or management interfaces to get unauthorized access to product assets via physical access in the default configuration of the product.

Risk factors	Likelihood	Security profiles
PHYS = 0	Low	LR, IoT-1
all others	Medium	IoT-2, IoT-3, RO-1, PC-*, PS-1, SE-*
$\text{PHYS} > 0$ & $\max(\text{SNDS}, \text{SNDT}, \text{SENF}) = 2$	High	OT-1, MOB-1, WE-1, LA-*

Risk factors	Impact	Security profiles
$\max(\text{SNDS}, \text{SNDT}, \text{SENF}) = 0$	Low	LR, IoT-1
$\max(\text{SNDS}, \text{SNDT}, \text{SENF}) = 1$	Medium	IoT-2, IoT-3
$\max(\text{SNDS}, \text{SNDT}, \text{SENF}) = 2$	High	RO-1, OT-1, MOB-1, WE-1, PC-*, LA-*, PS-1, SE-*

Requirements that mitigate this threat: SDEF, AUTH, LMAS, LOGG

Mitigations for Likelihood:

- Medium to Low: PDDI-1, AUTH
- High to Low: ADEF, PDDI-2, AUTH

Mitigations for Impact:

- Medium to Low: LOGG
- High to Low: JSTY, LOGG

C.4.6 TH-UAPS: Unauthorized access to product assets via unprotected local software access in default configuration

Attacker may use unprotected debug or management interfaces to get unauthorized access to product assets via local software access in the default configuration of the product.

Risk factors	Likelihood	Security profiles
$\max(\text{NUSR}, \text{SWMD}) = 0$	Low	LR, IoT-*,
all others	Medium	RO-1, OT-1, WE-1
$\max(\text{NUSR}, \text{SWMD}) = 2$ & $\max(\text{SNDS}, \text{SNDT}, \text{SENF}) = 2$	High	MOB-1, PC-*, LA-*, PS-1, SE-*

Risk factors	Impact	Security profiles
$\max(\text{SNDS}, \text{SNDT}, \text{SENF}) = 0$	Low	LR, IoT-1
$\max(\text{SNDS}, \text{SNDT}, \text{SENF}) = 1$	Medium	IoT-2, IoT-3
$\max(\text{SNDS}, \text{SNDT}, \text{SENF}) = 2$	High	RO-1, OT-1, MOB-1, WE-1, PC-*, LA-*, PS-1, SE-*

Requirements that mitigate this threat: SDEF, AUTH, LMAS, LOGG

Mitigations for Likelihood:

- Medium to Low: PDDI-1, AUTH
- High to Low: ADEF, PDDI-3, AUTH

Mitigations for Impact:

- Medium to Low: LOGG
- High to Low: JSTY, LOGG

C.4.7 TH-UAPN: Unauthorized access to product assets via unprotected network interfaces in default configuration

Attacker may use unprotected debug or management interfaces to get unauthorized access to product assets via the network in the default configuration of the product.

Risk factors	Likelihood	Security profiles
$\max(\text{FNET}, \text{TNET}) = 0$	Low	LR, IoT-1
all others	Medium	IoT-2, IoT-3
$\max(\text{FNET}, \text{TNET}) > 0$ & $\max(\text{SNDS}, \text{SNDT}, \text{SENF}) = 2$	High	RO-1, OT-1, MOB-1, WE-1, PC-*, LA-*, PS-1, SE-*

Risk factors	Impact	Security profiles
$\max(\text{SNDS}, \text{SNDT}, \text{SENF}) = 0$	Low	LR, IoT-1
$\max(\text{SNDS}, \text{SNDT}, \text{SENF}) = 1$	Medium	IoT-2, IoT-3
$\max(\text{SNDS}, \text{SNDT}, \text{SENF}) = 2$	High	RO-1, OT-1, MOB-1, WE-1, PC-*, LA-*, PS-1, SE-*

Requirements that mitigate this threat: SDEF, AUTH, LMAS, LOGG

Mitigations for Likelihood:

- Medium to Low: PDDI-1, AUTH
- High to Low: ADEF, PDDI-4, AUTH

Mitigations for Impact:

- Medium to Low: LOGG
- High to Low: JSTY, LOGG

C.4.9 TH-UADT: Unauthorized access to confidential data transmitted

Attacker may use network access to get unauthorized access to confidential data transmitted by the product.

Risk factors	Likelihood	Security profiles
TNET = 0	Low	LR, IoT-1
TNET = 1	Medium	IoT-2, IoT-3, OT-1, WE-1
TNET = 2	High	RO-1, MOB-1, PC-*, LA-*, PS-1, SE-*

Risk factors	Impact	Security profiles
SNDT = 0	Low	LR, IoT-1
SNDT = 1	Medium	IoT-2, IoT-3, OT-1
SNDT = 2	High	WE-1, RO-1, MOB-1, PC-*, LA-*, PS-1, SE-*

Requirements that mitigate this threat: CDTX, IDTX, DMIN

Mitigations for Likelihood:

- Medium to Low: DOCC
- High to Low: CDTX, DCTX, DOCC

Mitigations for Impact:

- Medium to Low: DJST
- High to Low: DJST

C.4.10 TH-PDOS: Denial of service attack on product functions via user or network access

Attacker may use user or network access for a denial-of-service attack on product functions.

Risk factors	Likelihood	
$\max(\text{NUSR}, \text{CUSR}, \text{FNET}) = 0$	Low	LR, IoT-1
$\max(\text{NUSR}, \text{CUSR}, \text{FNET}) = 1$	Medium	IoT-2, IoT-3, OT-1, MOB-1, WE-1, PC-*, LA-*, SE-1, SE-2
$\max(\text{NUSR}, \text{CUSR}, \text{FNET}) = 2$	High	RO-1, PS-1, SE-3

Risk factors	Impact	Security profiles
SENF = 0	Low	LR, IoT-1, IoT-2, WE-1
SENF = 1	Medium	IoT-3, PC-1, LA-1, PS-1
SENF = 2	High	RO-1, OT-1, MOB-1, PC-2, LA-2, SE-*

Requirements that mitigate this threat: AUTH, AVAI, LMII, LOGG, VULH

Mitigations for Likelihood:

- Medium to Low: DOST
- High to Low: DOST, VULH

Mitigations for Impact:

- Medium to Low: LMEM, LOGG
- High to Low: AUTH, AVNT, FDRP, LMEM, FAIR, LOGG

C.4.11 TH-DDOS: Denial of service attack on other products via exploitation of vulnerabilities or unauthorized use of product functions

Attacker may use the network to exploit vulnerabilities in the product to attack other products.

Guidance: Traffic amplification attacks and other misuses of product functions are considered vulnerabilities and/or unauthorized use for the purpose of this threat.

Risk factors	Likelihood	Security profiles
FNET = 0	Low	LR, IoT-1, WE-1
FNET = 1	Medium	IoT-2, IoT-3, OT-*, MOB-1, PC-*, LA-*
FNET = 2	High	RO-1, PS-1, SE-*

Risk factors	Impact	Security profiles
TNET = 0	Low	LR, IoT-1
TNET = 1	Medium	IoT-2, IoT-3, OT-*, WE-1
TNET = 2	High	RO-1, MOB-1, PC-*, LA-*, PS-1, SE-*

Requirements that mitigate this threat: NKEV, SSDD, SDEF, MSAF, LMII, MINI, LMAS, LOGG

All mitigations from TH-KEVU apply (using that requirement's risk formula), plus:

Mitigations for Impact:

- Medium to Low: (MDOC or MAMP)

- High to Low: MNET, MAMP

C.4.12 TH-MQSE: Masquerading authorized server

Attacker may masquerade as an authorized server to get unauthorized access to product assets.

Risk factors	Impact	Security profiles
TNET = 0	Low	LR, IoT-1
TNET = 1	Medium	IoT-2, IoT-3, OT-1, WE-1
TNET = 2	High	RO-1, MOB-1, PC-*, LA-*, PS-1, SE-*

Risk factors	Impact	Security profiles
max(SNDS, SNTD, SENF) = 0	Low	LR, IoT-1
max(SNDS, SNTD, SENF) = 1	Medium	IoT-2, IoT-3
max(SNDS, SNTD, SENF) = 2	High	RO-1, OT-1, MOB-1, WE-1, PC-*, LA-*, PS-1, SE-*

Requirements that mitigate this threat: CDTX, CRYP, IDTX, AUTH, LOGG

Mitigations for Likelihood:

- Medium to Low: AUTH, SUDC, CDTX, CRYP, IDTX
- High to Low: AUTH, SUDC, CDTX, CRYP, IDTX

Mitigations for Impact:

- Medium to Low: LOGG
- High to Low: LOGG

C.4.13 TH-LEAK: Data leak through side channels

Attacker may use the ability to run arbitrary software on the product to get unauthorized read access to confidential data.

Risk factors	Likelihood	Security profiles
CUSR = 0 or max(SNDS, SNTD) = 0	Low	LR, IoT-*, RO-1, OT-1, WE-1
all others	Medium	SE-1, PC-*, LA-*
CUSR = 2 & max(SNDS, SNTD) = 2	High	PS-1, SE-2, SE-3

Risk factors	Impact	Security profiles
max(SNDS, SNTD) = 0	Low	LR, IoT-1
max(SNDS, SNTD) = 1	Medium	IoT-2, IoT-3, WE-1
max(SNDS, SNTD) = 2	High	RO-1, OT-1, MOB-1, PC-*, LA-*, PS-1, SE-*

Requirements that mitigate this threat: MISO, DMIN, VULH

Mitigations for Likelihood:

- Medium to Low: VULH
- High to Low: (TRMD or PMSC), VULH

Mitigations for Impact:

- Medium to Low: DMIN
- High to Low: DMIN

C.5 Mapping of use cases to risk factors

NOTE: The “TOTAL” field is referenced by but does not define the security assurance level assignments table in Annex C.7.3 Table 1. It is primarily a consistency check to see if the risk factors sufficiently distinguish the differences in risk tolerance between use cases.

Use case	N U S R	C U S R	D A T A	P P I I	S N D S	S N D T	S E N F	P H Y S	U E I N	L O S S	H W M D	S W M D	D V C S	T N E T	F N E T	C O N F	A D M N	S U P P	T O T A L
UC-LR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
UC-IoT-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	2
UC-IoT-2	0	0	1	0	1	1	0	0	1	0	0	0	0	1	1	1	2	2	11
UC-IoT-3	0	0	1	0	1	1	1	0	2	0	1	0	0	1	1	1	2	2	14
UC-RO-1	0	0	1	0	1	2	2	0	2	0	0	0	0	2	2	2	1	2	17
UC-OT-1	0	0	0	0	1	1	2	2	1	0	0	0	0	1	1	2	1	2	14
UC-MOB-1	1	1	2	2	2	2	2	2	2	2	0	2	2	2	1	2	2	2	31
UC-WE-1	0	0	1	1	2	2	0	1	2	1	0	1	0	1	0	1	2	1	16
UC-PC-1	1	1	2	1	2	2	1	0	2	0	2	2	1	2	1	2	2	2	26
UC-PC-2	1	1	2	1	2	2	2	0	2	0	2	2	1	2	1	2	1	2	26
UC-LA-1	1	1	2	1	2	2	1	1	2	1	1	2	2	2	1	2	2	2	28
UC-LA-2	1	1	2	1	2	2	2	1	2	1	1	2	2	2	1	2	1	2	28
UC-PS-1	2	2	2	0	2	2	1	0	2	0	1	2	1	2	2	2	1	2	26
UC-SE-1	1	1	2	0	2	2	2	0	2	0	1	2	1	2	1	2	0	2	23
UC-SE-2	2	1	2	0	2	2	2	0	2	0	1	2	1	2	1	2	0	2	24
UC-SE-3	2	2	2	0	2	2	2	0	2	0	1	2	1	2	2	2	0	2	26

C.6 Security profiles and security assurance levels

C.6.1 General

Security profiles are an informative resource to the assessor. Each security profile is associated with a collection of levels of risk factors. Security profiles will be mapped to specific mitigations for each security requirements necessary to treat the risk.

C.6.2 Mapping of security profiles to risk factors

Security profiles are associated with sets of risk factor levels. Each security profile represents one or more use cases whose risks can be treated with the same set of mitigations.

Sec. Prof.	N U S R	C U S R	D A T A	P P I I	S N D S	S N D T	S E N F	P H Y S	U E I N	L O S S	H W M D	S W M D	D V C S	T N E T	F N E T	C O N F	A D M N	S U P P	T O T A L
SP-LR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SP-IoT-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	2
SP-IoT-2	0	0	1	0	1	1	0	0	1	0	0	0	0	1	1	1	2	2	11
SP-IoT-3	0	0	1	0	1	1	1	0	2	0	1	0	0	1	1	1	2	2	14
SP-RO-1	0	0	1	0	1	2	2	0	2	0	0	0	0	2	2	2	1	2	17
SP-OT-1	0	0	0	0	1	1	2	2	1	0	0	0	0	1	1	2	1	2	14
SP-MOB-1	1	1	2	2	2	2	2	2	2	2	0	2	2	2	1	2	2	2	31
SP-WE-1	0	0	1	1	2	2	0	1	2	1	0	1	0	1	0	1	2	1	16
SP-PC-1	1	1	2	1	2	2	1	0	2	0	2	2	1	2	1	2	2	2	26
SP-PC-2	1	1	2	1	2	2	2	0	2	0	2	2	1	2	1	2	1	2	26
SP-LA-1	1	1	2	1	2	2	1	1	2	1	1	2	2	2	1	2	2	2	28
SP-LA-2	1	1	2	1	2	2	2	1	2	1	1	2	2	2	1	2	1	2	28
SP-PS-1	2	2	2	0	2	2	1	0	2	0	1	2	1	2	2	2	1	2	26
SP-SE-1	1	1	2	0	2	2	2	0	2	0	1	2	1	2	1	2	0	2	23
SP-SE-2	2	1	2	0	2	2	2	0	2	0	1	2	1	2	1	2	0	2	24
SP-SE-3	2	2	2	0	2	2	2	0	2	0	1	2	1	2	2	2	0	2	26

C.6.3 Security assurance levels

Security assurance level (SAL) is a rough categorization of the level of security assurance a product provides if it meets the requirements associated with a security profile.

- **[SA-CRI]:** The product is suitable for use in highly sensitive or critical environments.
- **[SA-HIGH]:** The product is suitable for use in environments with low risk tolerances.
- **[SA-MED]:** The product is suitable for use in environments with medium risk tolerances. Some security tradeoffs may be made to improve usability and potential harms from unmitigated risk are widely acceptable.
- **[SA-LOW]:** The product is suitable for use in environments with high risk tolerances. Potential harms from unmitigated risks are low or negligible, and users are unlikely to reasonably expect security updates within the product's foreseeable use.

C.6.4 Mapping of security profile to security assurance level

Security assurance levels are informed by but not determined by the risk factor total from clause C.6.2.

Security profile	Description	RF total	SA L
SP-LR	Operating system for learning and research	0	LO W
SP-IoT-1	Non-internet-connected device such as a bluetooth speaker	2	LO W
SP-IoT-2	Internet-enabled power switch	11	ME D
SP-IoT-3	Internet-connected "smart home" device	14	ME D
SP-RO-1	Consumer-grade home wireless router	17	ME D
SP-OT-1	Business-grade remote door locking system	14	ME D
SP-MOB-1	Personal mobile device	31	HIG H
SP-WE-1	Wearable health tracker	16	ME D
SP-PC-1	Personal computer in a fixed and generally safe location	26	ME D
SP-PC-2	Enterprise workstation in a fixed and generally safe location	26	ME D
SP-LA-1	Personal laptop	28	HIG H
SP-LA-2	Enterprise laptop	28	HIG H
SP-PS-1	Personal server	26	ME D
SP-SE-1	Enterprise server in a datacenter with no user accounts	23	ME D
SP-SE-2	Enterprise server in a datacenter with only trusted user accounts	24	ME D
SP-SE-3	Enterprise server in a datacenter hosting many untrusted user accounts	26	ME D

Annex D (informative): Risk evaluation guidance

D.1 Explanation of Risk Modeling Approach

The risk modeling approach followed in this document can be applied to two situations:

1. *Covered*: For Manufacturers of products with use cases that are present in the text of this document, it states the mitigations which the product shall implement and provides guidance on how to verify that the mitigations are implemented in a product. Furthermore, it describes why that unique set of mitigations is sufficient for the use case.
2. *Not Covered*: For Manufacturers of products whose use case does not precisely match use cases covered in the present document, the methodology used herein may be further used to derive the appropriate set of mitigations for a given product, and to communicate this justification in a structured way. This could inform revisions of this document and the list of use cases over time.

Methodology

This clause describes the methodology followed in the current text.

1. Document a comprehensive range of foreseeable use cases for products of this type.
2. For a particular use case, document the inherent and product-specific risk factors likely to affect products of that type which are not already covered by other relevant standards.
3. For that use case, document environmental risk factors likely to affect products of that type which are not already covered by other relevant standards.
4. Document a comprehensive list of threats. For each threat, create a formula to estimate the risk level using the risk factors.
5. For each threat, document appropriate mitigations which should be present to mitigate the specific risk depending on the risk level. For each mitigation, also document at least one verification methodology.
6. Create a mapping between each use case and each risk factor, assigning a proportionality score. The scoring range should start from zero, representing the inapplicability of a risk factor to a use case, and increase monotonically based on both the likelihood and severity of potential harm or impact.
7. Develop security profiles from the use cases, which are collections of risk factor levels that can be used to fully describe the risk levels of all relevant threats. There may be one use case per security profile or multiple. There should be as many security profiles as are useful to manufacturers.
8. Using the risk factors in the security profiles and the risk formulas and mitigations for all threats, derive the completed list of required mitigations for each security profile.

D.2 Mapping of risks to requirements

Threat	Requirements
UEVU	SSDD, MSAF, LMII, LMAS, DMIN, LOGG
KEVU	NKEV, SSDD, MSAF, LMII, SCUD, LMAS, DMIN, AVAI, LOGG, VULH
CONF	CDST, SDEF, DMIN, LOGG
UADT	CDTX, DMIN
AVAI	AUTH, AVAI, LMAS, LOGG, VULH
PDOS	AUTH, AVAI, LMAS, LOGG, VULH
DDOS	AVAI, LMII, MINI, LMAS, LOGG, VULH
MQSE	CDTX, CRYP, IDTX, AUTH, SCUD, LOGG
LEAK	MISO, DMIN, VULH

D.3 Risk acceptance criteria

If the Likelihood and Impact of a risk are already Low or have been reduced to Low by application of mitigations, then the risk is acceptable. Alternatively, the risk may be transferred to the user or the operational environment, given proper justification.

D.4 Risks not treated by the requirements

For each risk untreated by the product itself, a corresponding mitigation has been created to explicitly permit the risk to be transferred to the user or operational environment. These are:

- MI-KEVD
- MI-KEVM
- MI-MDOC
- MI-PDDI-1
- MI-SUDC
- MI-SUOE
- MI-SUAO
- MI-SCHL
- MI-SCHM
- MI-SCHH
- MI-DOCC
- MI-DOST

Annex E (informative): Change history

The “Change history/Change request (history)” annex shall be included in every revised or amended harmonised standard and shall contain information concerning significant changes that have been introduced by it. It shall be presented as a table.

Date	Version	Information about changes
<Month year>	<#>	<Changes made are listed in this cell>

History

The following table will automatically be filled in by the ETSI Secretariat.

Document History		
Version	Date	Milestone
	<#>	