ALLIANCE FOR TELECOMMUNICATIONS INDUSTRY SOLUTIONS

ASSOCIATION OF HOME APPLIANCE MANUFACTURERS

BSA | THE SOFTWARE ALLIANCE

CABLELABS

COALITION FOR CYBERSECURITY POLICY AND LAW

COMPTIA

CONSUMER TECHNOLOGY ASSOCIATION

COUNCIL TO SECURE THE DIGITAL ECONOMY

CTIA

INDUSTRIAL INTERNET CONSORTIUM

INFORMATION TECHNOLOGY INDUSTRY COUNCIL

INTERNET OF SECURE THINGS

INTERNET SOCIETY

IOTOPIA

NCTA — THE INTERNET & TELEVISION ASSOCIATION

OPEN CONNECTIVITY FOUNDATION

TELECOMMUNICATIONS INDUSTRY ASSOCIATION

UL

U.S. CHAMBER OF COMMERCE

USTELECOM — THE BROADBAND ASSOCIATION

The C2 Consensus on IoT Device Security Baseline Capabilities

2021 SUPPLEMENT

CSDE

Council to Secure the Digital Economy

This document is a supplement to ***The C2 Consensus on IoT Device Security Baseline Capabilities***, published by the Council to Secure the Digital Economy (CSDE). The C2 Baseline and this Supplement are the result of the efforts of the many organizations that came together to produce the original consensus and continue to contribute today. In this document, the partners of the C2 Consensus update the Baseline with the results of 2020 and provide perspective on 2021. CSDE thanks the many contributors to this work as well as the legion of engineers, developers and scientists contributing to solutions to IoT security in the global digital economy.

CONTENTS

01 : Foreword

CSDE'S ORIGINAL CONVENE THE CONVENERS (C2) PROJECT was remarkable in two ways. First, the idea that a few standards bodies, trade associations and industry coalitions might be interested in cooperating on a joint consensus baseline generated a remarkable response. Twenty organizations participated, contributed and lent their name to the document; even more participated informally without accepting credit for their contributions.

The Consensus was also well received. In one year, it has been presented, cited and referenced in many security discussions. This common agreement between many industry organizations was deemed appropriate to various mappings between standards and regulatory requirements as everyone sought to find common ground in baseline discussions. APEC organizers invited a talk on the C2 Consensus at an IoT security conference in Malaysia, to help regional regulators and security experts understand how baselines can help secure the global IoT ecosystem. The cybersecurity Specialist Committee of ISO/IEC JTC1, SC27, used C2 as one of the inputs to building an international technical standard, currently 1st Working Draft 27402. And a technical standard that maps one-to-one with each specific Capability in the C2 Consensus was just published as CTA-2088, *Baseline Cybersecurity Standard for Devices and Device Systems*.

The strength of the original process was described in the Foreword of the original C2: "The convening—bringing together—of these groups allowed for sharing and comparing the expert recommendations each had developed within their own constituency."

This 2021 Supplement to the original 2019 *C2 Consensus on IoT Device Security Baseline Capabilities* reaffirms the primary Baseline guidance in the first Consensus and provides an update on related activities since the first publication. It discusses the status of items on the 2019 document's "roadmap", the Future Secure Capabilities discussed in Annex A of that document. It also updates the mappings listed in the extensive Annex D – Annex S portion of the Consensus.

02 : Updating the C2 Consensus—2020 In Review

This 2021 Supplement updates the *C2 Consensus on IoT Device Security Baseline Capabilities* ("C2 Consensus", Sep. 2019), a project hosted by the Council to Secure the Digital Economy (CSDE). CSDE convened the original C2 Consensus, but involved the participation of twenty major standards bodies, technical alliances, and civil society groups. The "C2" refers to the process of convening these entities, each of which convenes their own membership on cybersecurity topics—thus, "Convene the Convenors" or C2. The C2 Consensus leveraged these organizations' extensive cybersecurity expertise to achieve common agreement on a minimum set of connected device capabilities to support secure configuration, operation, maintenance, and end-of-life processes.

In this 2021 Guide update, we reaffirm the guidance on Baseline IoT device capabilities in the 2019 edition of the C2 Consensus. Rather than extend or change the list of capabilities, we have chosen to focus on issues of deployment and implementation of this guidance. This Supplement updates the mapping between the C2 Baseline and a number of other major regional and international standards, recommendations and regulatory actions. The *Future Capabilities* from the original C2 Consensus is updated here with progress by the National Cybersecurity Center of Excellence (U.S. Department of Commerce, National Institute for Standards and Technology) on device intent signaling and device onboarding. *The Informative References* section is updated and expanded.

Many contributed to this effort. We hope this guidance is helpful in navigating the challenging waters of cybersecurity for connected devices.

03 : Developments In IoT Baseline Security

Many stakeholders have been working to improve IoT security since the release of the C2 Consensus in 2019. This section will consider a few that are directly related.

RELEASE OF NISTIR 8259/8259A

An important event in 2020 was the publication of the NIST Core IoT Baseline and Guidance documents. The Core was released as NISTIR 8259A—IoT Device Cybersecurity Capability Core Baseline by the NIST Cybersecurity for IoT Program. This document lays out a "core" list of IoT device capabilities for manufacturers. The document also provides guidance on how to interpret these capabilities regarding the needs of industry sectors or individual manufacturers.

Released at the same time, NISTIR 8259—Foundational Cybersecurity Activities for IoT Device Manufacturers is guidance to manufacturers regarding activities recommended to address customer needs for cybersecurity. Both documents received significant attention and comments from stakeholders, including but not limited to industry, throughout their development.

Relationship of NISTIR 8259/8259A to the C2 Consensus

The Baseline in NISTIR 8259A represents a foundation upon which more industry-specific baselines and requirements may be built. The C2 Consensus is a *multi-sector core baseline* that extends and maps the six capabilities in 8259A to a broader set. Generally speaking, these device capabilities can be observed and verified, although implementation details—which are not part of 8259A—may make such observation or verification more difficult or less difficult in practice.

PUBLICATION OF A C2-BASED TECHNICAL STANDARD

The C2 Consensus lists ten baseline secure device capabilities and three baseline product lifecycle management activities. However, it is a guidance document and not written as a technical standard or with conformity assessment in mind. Software security architects, product managers, third party testers and others also need a detailed engineering standard with specific, testable assertions (requirements). The C2 Consensus fills an important role, but not the role of a technical standard.

In 2020, the Consumer Technology Association published a technical standard based entirely on the C2 Consensus, directly converting those ten capabilities and three activities to specific technical requirements and recommendations. CTA-2088, *Baseline Cybersecurity Standard for Devices and Device Systems*, extends the guidance into the realm of technical standards, an important step for product developers and assessors.

Relationship of CTA-2088 to the C2 Consensus

CTA-2088 was derived directly from the C2 Consensus. The 13 main capabilities sections of the C2 Consensus are replicated in CTA-2088, and the C2 definition of each capability is quoted at the beginning of the corresponding section of CTA-2088, to frame the capability properly in the same way as the C2 Consensus.

CTA-2088 is available from CTA at https://shop.cta.tech/collections/standards/cybersecurity.

UL IOT SECURITY RATING/UL MCV 1376 – SECURITY CAPABILITIES VERIFIED

In 2019, UL introduced the IoT Security Rating as a solution for manufacturers and buyers of connected devices to address baseline security. The UL IoT Security Rating features five levels (Bronze, Silver, Gold, Platinum and Diamond), ranging from minimum baseline to more comprehensive security capabilities. Devices can be assessed to any of these levels and obtain a differentiated product security rating with associated UL Verified Mark label.

The UL IoT Security Rating assesses and verifies security features of a device in line with and mapped to emerging industry consensus as covered in various baseline security best practices, frameworks and standards. These include but are not limited to the NIST Core Cybersecurity Feature Baseline (NISTIR 8259/8259A), the CSDE C2 Consensus on IoT Device Baseline Security, CTA-2088, the UK Code of Practice for Consumer IoT Security, and ETSI EN 303 645.

According to UL, their IoT Security Rating can help support manufacturers, distributors and retailers in demonstrating the threshold of "reasonable security features" in the California Bill for the Cybersecurity of Connected Devices (SB-327) and the Oregon House Bill 2395.

Relationship of the UL IoT Security Rating to the C2 Consensus

UL has defined its IoT Security Rating requirements, which are layered over five security levels and documented in "UL MCV 1376—Security Capabilities Verified", based on many of the same baseline security capabilities that are defined by the CSDE C2 Consensus.

For example, both the UL IoT Security Rating and the C2 Consensus reference the same essential security capabilities related to ensuring IoT devices can be securely updated, avoiding the use of (shared) default passwords across devices, ensuring data is protected both *at rest* and *in transit*, ensuring the use of industry-accepted cryptographic techniques, and requiring the vendor to follow a dedicated vulnerability management process.

The UL IoT Security Rating also leverages requirements reflected in a few other industry standards and frameworks. The majority of the C2 Consensus baseline security capabilities are referenced in the first two IoT Security Rating levels, Bronze and Silver, and the few remaining C2 Consensus baseline capabilities are found in two of the higher levels, Gold and Platinum. Therefore, the UL IoT Security Rating can serve as a means to demonstrate conformance to the CSDE C2 Consensus baseline security capabilities.

The UL IoT Security Rating (UL MCV 1376) is available from UL at https://ims.ul.com/IoT-security-rating.

04 : Review of Progress to Futures

In the C2 Annex *Regarding Future Secure Capabilities—Phase In Over Time*, the C2 working group deferred "baseline" status of two candidate capabilities because the technology was not deemed ready for broad deployment. Promising technologies did exist. But the group agreed that industry did not, in mid-2019, have adequate successful deployment experience to treat these items as baseline capabilities.

To be a baseline capability, a technology must be both necessary and feasible. Considering other capabilities in the baseline, such as "Device Identification" and "Cryptography", the science and practice are robust, well-documented, well-understood and routinely deployed.

However, the two candidates discussed here were clearly important for future consideration. We therefore update the status of the maturation and deployment of these technologies en route to a potential addition of them to the full C2 Baseline.

DEVICE INTENT SIGNALING

In September 2020, the U.S. Department of Commerce's National Institute of Standards and Technology released a study by the National Cybersecurity Center of Excellence: *Securing Small-Business and Home Internet of Things (IoT) Devices Mitigating Network-Based Attacks Using Manufacturer Usage Description (MUD)* ([NIST SP1800-15A] in Annex D: Informative References).

This study "demonstrated the practicality and effectiveness of using the Internet Engineering Task Force's Manufacturer Usage Description (MUD) standard to reduce both the vulnerability of IoT devices to network-based attacks and the potential for harm from any IoT devices that become compromised."[1] Eleven technology partners and collaborators participated with NCCoE.

The project demonstrated a working system that constrained device behavior (prevented devices from accessing non-whitelisted internet resources) using RFC 8520, the IETF MUD standard. The multi-volume guide includes a how-to section for implementers.

Further information is available in this comprehensive study report and implementation guide [NIST SP1800-15A].

DEVICE NETWORK ONBOARDING

The same NCCoE project as described above also demonstrated secure and automated onboarding of both MUD-capable and non-MUD-capable devices using 423 thwe Wi-Fi Alliance's Wi-Fi Easy Connect protocol in a Micronets Mobile App, in Build 3 of the system.

As described in the report, "Using the Wi-Fi Easy Connect protocol to onboard devices ensures that there is no need for anyone to be privy to the device's network credentials. The onboarding protocol provisions the network

credentials onto the device automatically, using a secure channel, and the device is then able to present its credentials to the network as part of the standard Wi-Fi network connection handshake. There is no need for the device's network password to be input by a human, and the credentials are never displayed, so presentation of the device's network credentials to the network does not pose any risk that the credentials will be viewed and thereby disclosed."[2]

Please see the full guide [NIST SP1800-15A] for more information.

05 : Annex A: Informative References

The work of the C2 Consensus organizations draws on recommendations by these groups and others. The following references are informative. Where the reference is used elsewhere in this document, it is denoted by a reference tag in square brackets ("[ ]").

- [BSA] BSA | The Software Alliance, *"BSA Framework for Secure Software"*, https://www.bsa.org/files/reports/bsa_framework_secure_software_update_2020.pdf
- [CSDE] Council to Secure the Digital Economy (CSDE), *"International Botnet and IoT Security Guide"*, November 2020, https://CSDE.tech/projects/international-anti-botnet-guide/
- [CTIA IoT CC] CTIA, *Cybersecurity Certification Test Plan for IoT Devices*, October 2018, https://ctiacertification.org/program/iot-cybersecurity-certification/
- [DCMS] United Kingdom Department for Digital, Culture, Media and Sport (UK DCMS), *Code of Practice for consumer IoT security*, October 2018, https://www.gov.uk/government/publications/code-of-practice-for-consumer-iot-security/code-of-practice-for-consumer-iot-security
- [EN303645] *Cyber Security for Consumer Internet of Things: Baseline Requirements*, https://www.etsi.org/deliver/etsi_en/303600_303699/303645/02.01.00_30/en_303645v020100v.pdf
- [ENISA] European Union Agency for Network and Information Security (ENISA), *Baseline Security Recommendations for IoT*, November 2017, https://www.enisa.europa.eu/publications/baseline-security-recommendations-for-iot
- [GSMA] Global System for Mobile Communications Association (GSMA), *GSMA IoT Security Guidelines for Endpoint Ecosystems*, February 2016, https://www.gsma.com/iot/wp-content/uploads/2016/02/CLP.13-v1.0.pdf
- International Society of Automation (ISA)/International Electrotechnical Commission (IEC) – 62443 series of standards on the cyber security of industrial automation and control systems, https://www.isa.org/isa99/
- [NIST SP1800-15A] National Institute of Standards and Technology (NIST, United States Department of Commerce),NIST SP1800-15A, *Securing Small-Business and Home Internet of Things (IoT) Devices Mitigating Network-Based Attacks Using Manufacturer Usage Description (MUD)*, September 2020, https://www.nccoe.nist.gov/projects/building-blocks/mitigating-iot-based-ddos
- [NISTIR 8228 Considerations] NISTIR 8228, *Considerations for Managing Internet of Things (IoT) Cybersecurity and Privacy Risks*, June 2019, https://doi.org/10.6028/NIST.IR.8228
- [NISTIR 8259 Activities] NIST, NISTIR 8259, *Foundational Cybersecurity Activities for IoT Device Manufacturers*, May 2020, https://doi.org/10.6028/NIST.IR.8259
- [NISTIR 8259A Baseline] NIST, NISTIR 8259A, *IoT Device Cybersecurity Capability Core Baseline*, May 2020, https://doi.org/10.6028/NIST.IR.8259A

- [OCF Security Specification ISO/IEC 30118-2:2018] Open Connectivity Foundation (OCF), *OCF Security Specification 2.0.1*, February 2019, https://www.iso.org/standard/74239.html (https://openconnectivity.org/specs/OCF_Security_Specification_v2.0.1.pdf)
- [OCF Security Specification] Open Connectivity Foundation (OCF), *OCF Security Specification 2.2.0*, July, 2020, https://openconnectivity.org/specs/OCF_Security_Specification_v2.2.0.pdf
- [OCF Wi-Fi Easy Setup Specification] Open Connectivity Foundation (OCF), *OCF Wi-Fi Easy Setup Specification 2.2.0*, July, 2020, https://openconnectivity.org/specs/OCF_Security_Specification_v2.2.0.pdf
- [Security Pledge] Internet of Secure Things (IoXT), *The IoXT Security Pledge*, https://www.ioxtalliance.org/s/ioXt-SecurityPledge-booklet-final.pdf
- [TS103645] European Telecommunications Standards Institute (ETSI), T*S 103 645 Cyber Security for Consumer Internet of Things*, February 2019, https://www.etsi.org/deliver/etsi_ts/103600_103699/103645/01.01.01_60/ts_103645v010101p.pdf
- [UL] UL, UL MCV 1376—Security Capabilities Verified, https://shopulstandards.com/ProductDetail.aspx?UniqueKey=35953
- [WOT BP] World Wide Web Coalition (W3C), *WoT Security Best Practices*, retrieved October 2020, https://w3c.github.io/wot-security-best-practices/
- [WOT TP] World Wide Web Coalition (W3C), *WoT Security Testing Plan*, retrieved October 2020, https://w3c.github.io/wot-security-testing-plan

06 : Annex B: Mapping to CSDE International Anti-Botnet Guide

The BSA *Framework for Secure Software* is an outcome-based, technology-neutral tool for understanding and evaluating security in software products and services, including software and software components used in IoT devices. Its mapping to the C2 document serves to identify best practices and informative references relating to implementing C2 practices in the software context. The mapping includes references to the BSA Framework in the following format: **function.category-subcategory**.

CATEGORY	MAPS TO
Secure Device Capabilities – Baseline DEVICE IDENTIFIERS	
Secure Device Capabilities – Baseline SECURED ACCESS	[BSA] **SI.1.** The software avoids architectural weaknesses that create risk of authentication failure. ▪ **SI.1-1.** The software avoids hardcoded passwords. ▪ **SI.1-2.** Software source code does not contain secrets. ▪ **SI.1-3.** Authentication mechanisms used by the software employ industry standard security techniques and avoid common security weaknesses. ▪ **SI.1-4.** The software does not store sensitive authentication information, which may include passwords or keys, in source code or publicly accessible infrastructure. ▪ **SI.1-5.** Any passwords or sensitive authentication information stored by the software is stored in accordance with current best practices. [BSA] **SI.2.** The software supports strong identity management and authentication. ▪ **SI.2-1.** The software implements features, configurations, and protocols that establish or support standard, tested authentication services. ▪ **SI.2-2.** The software is interoperable with applicable common industry standards for identity management and authentication. ▪ **SI.2-3.** Authentication controls fail securely.
Secure Device Capabilities – Baseline DATA IN TRANSIT IS PROTECTED	[BSA] **CS.1.** Software is developed in accordance with an encryption strategy that defines what data should be encrypted and which encryption mechanisms should be used. ▪ **CS.1-1.** Software enables the use of encryption to protect sensitive data from unauthorized disclosure or modification.
Secure Device Capabilities – Baseline DATA AT REST IS PROTECTED	[BSA] **CS.1.** Software is developed in accordance with an encryption strategy that defines what data should be encrypted and which encryption mechanisms should be used. ▪ **CS.1-1.** Software enables the use of encryption to protect sensitive data from unauthorized disclosure or modification.
Secure Device Capabilities – Baseline INDUSTRY ACCEPTED PROTOCOLS ARE USED FOR COMMUNICATIONS	
Secure Device Capabilities – Baseline DATA VALIDATION	[BSA] **SC.3-2.** Software validates input and output to mitigate common vulnerabilities in software.

CATEGORY	MAPS TO
Secure Device Capabilities – Baseline EVENT LOGGING	[BSA] **LO.1.** Software implements logging of all critical security incident and event information. ▪ **LO.1-1.** Software differentiates between monitoring logs and auditing logs. ▪ **LO.1-2.** Software is capable of logging all security-relevant failures, errors, and exceptions. ▪ **LO.1-3.** Software is capable of logging timestamp and identifying information associated with security incidents and events.
Secure Device Capabilities – Baseline CRYPTOGRAPHY	[BSA] **CS.2.** Software avoids weak encryption. ▪ **CS.2-1.** Software avoids custom encryption algorithms and implementations. ▪ **CS.2-2.** Software enables the use of authenticated encryption. ▪ **CS.2-3.** Cryptography employed by the software enables strong algorithms. ▪ **CS.2-4.** Cryptography employed by the software enables strong key lengths. ▪ **CS.2-5.** Encryption capabilities employed by the software are configured to select strong cipher modes and exclude weak ciphers by default. ▪ **CS.2-6.** Software is configured to disable or prevent the use of weak encryption algorithms and key lengths.
Secure Device Capabilities – Baseline PATCHABILITY	[BSA] **PA.1.** Software is capable of receiving secure updates and security patches. ▪ **PA.1-1.** Software is capable of validating the integrity of a transmitted patch or update. ▪ **PA.1-2.** Software includes a mechanism to notify end users of patch or update installation. ▪ **PA.1-3.** Software reverts to a known-good state upon failed installation of updates or security patches.
Secure Device Capabilities – Baseline REPROVISIONING	[BSA] **CF.1-6.** Software configuration settings can be altered to tailor security settings to the operating environment.
Product Lifecycle Management VULNERABILITY SUBMISSION AND HANDLING PROCESS	[BSA] **VM.1.** The vendor maintains an up-to-date vulnerability management plan. [BSA] **VM.2.** Vulnerabilities are identified and resolved rapidly and comprehensively, according to risk-based prioritization. [BSA] **VM.3.** The vendor maintains a coordinated vulnerability disclosure program. ▪ **VM.3-1.** The vendor establishes a clearly defined and easily accessible intake mechanism to accept vulnerability information (email, portal, etc.). ▪ **VM.3-2.** A vendor's intake mechanism provides for secure and confidential communication of sensitive vulnerability information. ▪ **VM.3-3.** The vendor publishes, in simple and clear language, its policies for interacting with vulnerability reporters, addressing, at minimum: (1) how the vendor would like to be contacted, (2) options for secure communication, (3) expectations for communication from the vendor regarding the status of a reported vulnerability, (4) desired information regarding a potential vulnerability, (5) issues that are out of scope of the vulnerability disclosure program, (6) how submitted vulnerability reports are tracked, and (7) expectations for whether and how a reporter will be credited. ▪ **VM.3-4.** The vendor maintains a system to record and track all reports of potential vulnerabilities. ▪ **VM.3-5.** The vendor notifies vulnerability reporters of when reported vulnerabilities are remediated or mitigated.

CATEGORY	MAPS TO
Product Lifecycle Management EOL/EOS UPDATES AND DISCLOSURE	[BSA] **EL.1.** Vendor maintains consistent lifecycle guidance. ▪ **EL.1-1.** Vendor communicates realistic assumptions and expectations regarding the nature and lifespan of product support in tandem with initial software delivery. ▪ **EL.1-2.** Vendor clearly communicates decisions to terminate support for a software product to customers and users, identifying the expected support termination date; the anticipated risk of continued product use beyond the termination of support; possible mitigation actions; and options for technical migration to replacement products. ▪ **EL.1-3.** Software is continually monitored to ensure that third-party components have not reached end-of-life milestones or are removed or otherwise remediated.
Product Lifecycle Management DEVICE INTENT DOCUMENTATION	
Secure Capabilities – Phase In Over Time DEVICE INTENT SIGNALING	
Secure Capabilities – Phase In Over Time DEVICE NETWORK ONBOARDING	
Additional IoT Device Security Capabilities and Practices SECURE DEVELOPMENT LIFECYCLE	The *BSA Framework for Secure Software* encapsulates secure development lifecycle practices throughout its three functions. The "Secure Development" function addresses security in the phase of software development when a software project is conceived, initiated, developed, and brought to market. The "Secure Capabilities" function identifies key security characteristics recommended for a software product. Finally, the "Secure Lifecycle" function addresses considerations for maintaining security in a software product from its development through the end of its life. The Framework in its entirety is recommended to inform understanding and evaluation of the Secure Development Lifecycle practice. See [BSA] for additional detail.
Additional IoT Device Security Capabilities and Practices HARDWARE ROOTED SECURITY	
Additional IoT Device Security Capabilities and Practices TIME DISTRIBUTION	
Additional IoT Device Security Capabilities and Practices SYSTEM RESILIENCY	

CATEGORY	MAPS TO
Additional IoT Device Security Capabilities and Practices SECURE TOOLCHAINS	[BSA] **DE.2.** Software is developed using tools configured for security. ▪ **DE.2-1.** Software is developed using up-to-date versions of all tools and platform elements within the development environment. ▪ **DE.2-2.** Development frameworks used in developing software use secure configurations. ▪ **DE.2-3.** Compilers are configured to prevent common vulnerabilities and weaknesses. ▪ **DE.2-4.** Compilers are configured to avoid unintentional removal or modification of security-critical code. ▪ **DE.2-5.** Compilers are configured to automatically add defense code. ▪ **DE.2-6.** Containers and other virtualization technologies used in deploying the software use secure configurations.
Additional IoT Device Security Capabilities and Practices SOFTWARE TRANSPARENCY AND BILL OF MATERIALS	[BSA] **SM.2**. Approved acquisition measures are in place to ensure the visibility, traceability, and security of third-party components. ▪ **SM.2-1.** Information about providers of third-party components is identified and collected. ▪ **SM.2-2.** Software development organization employs measures to document and, to the extent feasible, trace to their original source all third-party components directly acquired and incorporated into the software by the developer. ▪ **SM.2-3.** To the maximum feasible through the use of manual and automated technologies, subcomponents integrated in third-party components are documented, and their lineage and dependencies traced.
Additional IoT Device Security Capabilities and Practices LEAST FUNCTIONALITY	
Additional IoT Device Security Capabilities and Practices PHYSICAL ACCESS CONTROL	
Additional IoT Device Security Capabilities and Practices BEST CURRENT PRACTICES	The best practices outlined in the *BSA Framework for Secure Software* reflect the consensus of BSA members, the world's leading enterprise software developers, and are mapped throughout to widely recognized standards and other informative references. See [BSA] for additional detail.

07 : Annex C: Mapping to CSDE International Botnet and IoT Security Guide

Since the original release (as the International Anti-Botnet Guide 2019), the Guide IoT device guidance has been updated to match the C2 Consensus more closely.

CATEGORY	MAPS TO
Secure Device Capabilities – Baseline DEVICE IDENTIFIERS	[CSDE] 5.C.2.a: The device should have a unique value associated with it that is distinct and distinguishes the device from all other devices.
Secure Device Capabilities – Baseline SECURED ACCESS	[CSDE] 5.C.2.b: The device must be carefully protected by requiring user authentication to read or modify the software, firmware and configuration, including means to ensure device-unique credentials for administrative access, and by protecting access to interfaces.
Secure Device Capabilities – Baseline DATA IN TRANSIT IS PROTECTED	[CSDE] 5.C.2.c: The confidentiality and integrity of data at rest and in transit should be protected. To that end, data communications should be encrypted except in cases where risk analysis indicates otherwise. Sensitive data should be stored encrypted. In general, the security mechanisms available in whatever system is used should be employed to protect data at rest and in transit.
Secure Device Capabilities – Baseline DATA AT REST IS PROTECTED	[CSDE] 5.C.2.c: The confidentiality and integrity of data at rest and in transit should be protected. To that end, data communications should be encrypted except in cases where risk analysis indicates otherwise. Sensitive data should be stored encrypted. In general, the security mechanisms available in whatever system is used should be employed to protect data at rest and in transit.
Secure Device Capabilities – Baseline INDUSTRY ACCEPTED PROTOCOLS ARE USED FOR COMMUNICATIONS	[CSDE] 5.C.2.d: Use of secure, widely used protocols, excluding deprecated and replaced versions and protocols, for communications to and from the device.
Secure Device Capabilities – Baseline DATA VALIDATION	[CSDE] 5.C.2.e: Any input received from outside the system must be managed so that an outside adversary cannot arrange for it to be used directly as code, commands, or other execution flow inputs. Input should be validated for length, character type, and acceptable values or ranges. Output from one subsystem to another or to another site should also be filtered.
Secure Device Capabilities – Baseline EVENT LOGGING	[CSDE] 5.C.2.f: Relevant cybersecurity events should be recorded (subject to available memory space), secured and available to authorized users. Relevant events are application-specific, but examples include failed login attempts or negative results from cybersecurity checks such as boot time measurement or hash verification.
Secure Device Capabilities – Baseline CRYPTOGRAPHY	[CSDE] 5.C.2.g: Where cryptographic methods are used to ensure data integrity and confidentiality, rights authentication and non-repudiation of requests, they should be chosen to match the assessed risk. The implementation should use open, published, proven, and peer-reviewed cryptographic methods with appropriate parameter, algorithm and option selections. Where feasible, cryptographic methods should be updateable. Deprecated methods are to be avoided. Hardware-rooted security should be considered as to how it fits into the secure development lifecycles of current and future products. Device manufacturers should not rely solely on use of obfuscation to secure secrets (e.g., device keys, sensitive data), but obfuscation may be used to increase the difficulty of an attacker to locate the secret. Still, the secret should be protected by other means such as access control and encryption.

CATEGORY	MAPS TO
Secure Device Capabilities – Baseline PATCHABILITY	[CSDE] 5.C.2.h: A plan for secure updates with anti-rollback protection and proper access control throughout a defined security support period, where technically feasible.
Secure Device Capabilities – Baseline REPROVISIONING	[CSDE] 5.C.2.i: The manufacturer provides authorized users with the capability to securely reconfigure and redeploy a device post-market, especially to return the product to factory defaults or an authorized restore point, and securely remove data collected by the device (that is not essential to its operation), within a defined period established by the organization.
Product Lifecycle Management VULNERABILITY SUBMISSION AND HANDLING PROCESS	[CSDE] 5.C.3.a: Providers — manufacturers and retailers — should create a security vulnerability policy and process to identify, prioritize, mitigate, and where appropriate disclose known security vulnerabilities in their products.
Product Lifecycle Management EOL/EOS UPDATES AND DISCLOSURE	[CSDE] 5.C.3.b: Device providers should have a defined security support policy that includes the handling of any the end-of-life (EoL) or end-of-service (EoS) security vulnerabilities, whether updates will be made available and how, and what to do with the device at that time.
Product Lifecycle Management DEVICE INTENT DOCUMENTATION	[CSDE] 5.C.3.c: The device manufacturer provides documentation of the device's as-designed network usage publicly, either in product documentation or other means for device users.
Secure Capabilities – Phase In Over Time DEVICE INTENT SIGNALING	[CSDE] 5.C.2.j: The device supports the process of authenticating the device, authorizing it with credentials, and configuring it to communicate within the appropriate security domain. ("Advanced Capability")
Secure Capabilities – Phase In Over Time DEVICE NETWORK ONBOARDING	[CSDE] 5.C.2.k: The device supports a protocol for the device to provide information to routers or firewalls upstream regarding the intended network usage. Equivalently, the device provides heuristics related to its own behavior in normal operation in support of network analysis. ("Advanced Capability")
Additional IoT Device Security Capabilities and Practices SECURE DEVELOPMENT LIFECYCLE	[CSDE] 5.C.1.a: A secure development lifecycle (SDL) process should be in place. While specific elements of an SDL may vary, SDLs should include the following security-oriented elements: threat identification and disposition; coding standards; 3rd party software requirements; software security controls and capabilities test and validation; and new vulnerability identification and handling.
Additional IoT Device Security Capabilities and Practices HARDWARE ROOTED SECURITY	[CSDE] 5.C.2.g: Hardware-rooted security should be considered as to how it fits into the secure development lifecycles of current and future products.
Additional IoT Device Security Capabilities and Practices TIME DISTRIBUTION	

CATEGORY	MAPS TO
Additional IoT Device Security Capabilities and Practices SYSTEM RESILIENCY	
Additional IoT Device Security Capabilities and Practices SECURE TOOLCHAINS	[CSDE] 5.C.1.b: Tools that are able to check if the implementation is following secure coding guidelines and to search for a subset of known Common Vulnerabilities and Exposures (CVEs) should be used to develop, compile, build and maintain software. Memory-safe languages should also be used.
Additional IoT Device Security Capabilities and Practices: SOFTWARE TRANSPARENCY AND BILL OF MATERIALS	
Additional IoT Device Security Capabilities and Practices LEAST FUNCTIONALITY	
Additional IoT Device Security Capabilities and Practices PHYSICAL ACCESS CONTROL	
Additional IoT Device Security Capabilities and Practices BEST CURRENT PRACTICES	

08 : Annex D: Mapping to CTA-2088 *Baseline Cybersecurity Standard for Devices and Device Systems*

The original CTA-2088 technical standard was conceived as a basic standard for minimum capabilities, a current topic for IoT cybersecurity in 2018. The project began in late 2018 but was put on hold during the C2 Consensus work of 2019. When the project restarted, it was reframed as a direct 1:1 mapping of the guidance in the C2 Consensus to specific technical requirements that would implement that guidance. As a result, the mapping below is simple: Each of the 13 top-line elements in the C2 Consensus baseline sections 5 and 6 maps directly to an equivalent section in CTA-2088.

CATEGORY	MAPS TO
Secure Device Capabilities – Baseline DEVICE IDENTIFIERS	[CTA-2088] 5.1 Device Identifiers
Secure Device Capabilities – Baseline SECURED ACCESS	[CTA-2088] 5.2 Secured Access
Secure Device Capabilities – Baseline DATA IN TRANSIT IS PROTECTED	[CTA-2088] 5.3 Data In Transit Is Protected
Secure Device Capabilities – Baseline DATA AT REST IS PROTECTED	[CTA-2088] 5.4 Data At Rest Is Protected
Secure Device Capabilities – Baseline INDUSTRY ACCEPTED PROTOCOLS ARE USED FOR COMMUNICATIONS	[CTA-2088] 5.5 Industry Accepted Protocols are Used for Communications
Secure Device Capabilities – Baseline DATA VALIDATION	[CTA-2088] 5.6 Data Validation
Secure Device Capabilities – Baseline EVENT LOGGING	[CTA-2088] 5.7 Event Logging
Secure Device Capabilities – Baseline CRYPTOGRAPHY	[CTA-2088] 5.8 Cryptography
Secure Device Capabilities – Baseline PATCHABILITY	[CTA-2088] 5.9 Patchability
Secure Device Capabilities – Baseline REPROVISIONING	[CTA-2088] 5.10 Reprovisioning
Product Lifecycle Management VULNERABILITY SUBMISSION AND HANDLING PROCESS	[CTA-2088] 6.1 Vulnerability Submission and Handling Process

CATEGORY	MAPS TO
Product Lifecycle Management EOL/EOS UPDATES AND DISCLOSURE	[CTA-2088] 6.2 EoL/EoS Updates and Disclosure
Product Lifecycle Management DEVICE INTENT DOCUMENTATION	[CTA-2088] 6.3 Device Intent Documentation
Secure Capabilities – Phase In Over Time DEVICE INTENT SIGNALING	[CTA-2088] 8.1 Device Intent Signaling
Secure Capabilities – Phase In Over Time DEVICE NETWORK ONBOARDING	[CTA-2088] 8.2 Device Network Onboarding
Additional IoT Device Security Capabilities and Practices SECURE DEVELOPMENT LIFECYCLE	
Additional IoT Device Security Capabilities and Practices HARDWARE ROOTED SECURITY	
Additional IoT Device Security Capabilities and Practices TIME DISTRIBUTION	
Additional IoT Device Security Capabilities and Practices SYSTEM RESILIENCY	
Additional IoT Device Security Capabilities and Practices SECURE TOOLCHAINS	
Additional IoT Device Security Capabilities and Practices SOFTWARE TRANSPARENCY AND BILL OF MATERIALS	
Additional IoT Device Security Capabilities and Practices LEAST FUNCTIONALITY	
Additional IoT Device Security Capabilities and Practices PHYSICAL ACCESS CONTROL	
Additional IoT Device Security Capabilities and Practices BEST CURRENT PRACTICES	

09 : Annex E: Mapping to CTIA IoT Device Cybersecurity Certification

CATEGORY	MAPS TO
Secure Device Capabilities – Baseline DEVICE IDENTIFIERS	[CTIA IoT CC] 4.13 Device Identity is globally unique and required. Additional network components like a SIM/eSIM and MAC address are additional to the Globally Unique ID requirement. Additionally, device must provide its globally unique identity in the audit log
Secure Device Capabilities – Baseline SECURED ACCESS	CTIA IoT CC] 3.2: Password Management Test - Unique Default Password for each device Password Change required upon first login Password is of sufficient complexity and length [CTIA IoT CC] 3.3: Authentication Test - Authentication required to modify device settings [CTIA IoT CC] 3.4: Access Controls - Role Based Access Controls [CTIA IoT CC] 4.2: Password Management Test - Idle logout Password Integration with Enterprise Management System [CTIA IoT CC] 4.3: Access Control - Integrated password with Enterprise Management System [CTIA IoT CC] 4.9: Multi-factor Authentication - Multi-Factor Authentication is supported [CTIA IoT CC] 5.17 Designed In Feature - All Network Communications except those minimally required to function are disabled by default
Secure Device Capabilities – Baseline DATA IN TRANSIT IS PROTECTED	[CTIA IoT CC] 4.8 Encryption of Data in Transit - Required support for TLS, DTLS, SSH or IPSec for end to end encryption at minimal 128-bit AES. [CTIA IoT CC] 5.15 - Encryption of Data at Rest - Required support for encryption of data at rest at minimal 128-bit AES
Secure Device Capabilities – Baseline DATA AT REST IS PROTECTED	
Secure Device Capabilities – Baseline INDUSTRY ACCEPTED PROTOCOLS ARE USED FOR COMMUNICATIONS	[CTIA IoT CC] - CTIA recommends common peer reviewed industry standards Encryption in transit supports IPSEC, SSH, TLS and DTLS at the 128-bit AES support Encryption at Rest supports minimal 128-bit AES support Digital Signature Generation and Validation support RSA or ECDSA algorithms for X.509 certificates in P7S formats
Secure Device Capabilities – Baseline DATA VALIDATION	[CTIA IoT CC] 3.2 - validates inputs for password [CTIA IoT CC] 3.5/3.6 - validates patches and upgrades [CTIA IoT CC] 5.13 - validates digital certificates [CTIA IoT CC] 5.17 - validates network services minimally required
Secure Device Capabilities – Baseline EVENT LOGGING	[CTIA IoT CC] 4.7 Audit Log - Devices are required to handle 4 specific audit log type entries based on Syslog format. The four are emergency, alert, critical, and error audit log entries.

CATEGORY	MAPS TO
Secure Device Capabilities – Baseline CRYPTOGRAPHY	[CTIA IoT CC] 4.8 Encryption in Transit support minimally the 128-bit AES standard to protect data and support compatibility with the rest of IT ecosystem. It also supports strong, industry vetted protocols for end-to-end encryptions such as SSH, TLS, DTLS, and IPSec [CTIA IoT CC] 5.14 Digital Signature Validation and Generation supports industry adopted standards such as the RSA and the ECDSA algorithms to support strong X.509 certificates in P7S format. This protects software and supports strong authentication [CTIA IoT CC] 5.15 Encryption at Rest support minimally the 128-bit AES standard to protect data at rest and support compatibility with the rest of the IT ecosystem.
Secure Device Capabilities – Baseline PATCHABILITY	[CTIA IoT CC] 3.5 & 3.6, 4.5 & 4.6 Patches and Upgrades are a required element that is available at the lowest level from the manufacturer or at the managed level, provided by the managing enterprise infrastructure
Secure Device Capabilities – Baseline REPROVISIONING	
Product Lifecycle Management VULNERABILITY SUBMISSION AND HANDLING PROCESS	[CTIA IoT CC] 3.1 Terms of Service and Privacy Policy - Manufacturers state how long a device will be support for patches and upgrades that will address vulnerability handling at the device level.
Product Lifecycle Management EOL/EOS UPDATES AND DISCLOSURE	
Product Lifecycle Management DEVICE INTENT DOCUMENTATION	
Secure Capabilities – Phase In Over Time DEVICE INTENT SIGNALING	
Secure Capabilities – Phase In Over Time DEVICE NETWORK ONBOARDING	[CTIA IoT CC] This is covered by most of section 4 in the plan regarding connecting the device to an enterprise management system. For cellular based devices, there is also a requirement to get the device provision through the operator.
Additional IoT Device Security Capabilities and Practices SECURE DEVELOPMENT LIFECYCLE	
Additional IoT Device Security Capabilities and Practices HARDWARE ROOTED SECURITY	[CTIA IoT CC] 4.11 Secure Boot may be accomplished with the use of a hardware root of security such as a TPM module [CTIA IoT CC] 5.14 Digital Signature Generation and Validation may have a hardware root of trust module to support this functionality

CATEGORY	MAPS TO
Additional IoT Device Security Capabilities and Practices TIME DISTRIBUTION	
Additional IoT Device Security Capabilities and Practices SYSTEM RESILIENCY	
Additional IoT Device Security Capabilities and Practices SECURE TOOLCHAINS	Suggest UL CAP program for this activity
Additional IoT Device Security Capabilities and Practices SOFTWARE TRANSPARENCY AND BILL OF MATERIALS	
Additional IoT Device Security Capabilities and Practices LEAST FUNCTIONALITY	[CTIA IoT CC] 5.17 - Designed-In Features - One requirement is that the device separate critical from non-critical functions. Another requirement is that the device fail in a secure manner.
Additional IoT Device Security Capabilities and Practices PHYSICAL ACCESS CONTROL	[CTIA IoT CC] 5.16 Tamper Evidence - Devices at the CTIA Level 3 usually have secured if not hardened and weather rated enclosures meant to protect the device from case intrusion. As such, tamper evidence provides for silent notification if a case is opened and notification can be sent back to the network controllers
Additional IoT Device Security Capabilities and Practices BEST CURRENT PRACTICES	

10 : Annex F: Mapping to IoTopia Specifications

CATEGORY	MAPS TO
Secure Device Capabilities – Baseline DEVICE IDENTIFIERS	*Certificate based authentication. Onboarding requires a voucher with dev ID. MUD URL imbedded in device by manufacturer.* a. Endpoints that communicate via IEEE 802 networking must contain a certificate (IDevID) along with the MUD-URL, and associated private key for the certificate. [IEEE802.1AR] b. Heuristics: Manufacturers must provide a description of device behavior that may be used by the network to infer identities c. Endpoints that implement via IEEE 802 networking must support installation of at least one local certificate (LDevIDs) and associated private keying material.
Secure Device Capabilities – Baseline SECURED ACCESS	*"Device must utilize secure standard protocols and security mechanisms to provide multi-factor authentication for remote and local (physical) access to device* *a. Devices should not be able to support full operation with default passwords* *b. secure password enforcement should be imbedded in device* *c. as appropriate, passwords will require updates"* Prior to completing Onboarding (e.g. obtaining a local trust anchor and LDevID) Endpoints communicating on IEEE 802 networks MUST authenticate using their IDevID and must accept the local 802.1X network credentials without validation purely for the purposes of onboarding.
Secure Device Capabilities – Baseline DATA IN TRANSIT IS PROTECTED	*Secure boot, secure data storage, measured boot, voucher storage, key storage, crypto support, crypto upgrade potential* Endpoints must protect personally identifiable information from disclosure and modification. The actual implementation will depend on the nature of the endpoint and associated service, but an example would be to encrypt information on board the device such that only authorized users may access it.
Secure Device Capabilities – Baseline DATA AT REST IS PROTECTED	*Device manufacturer should provide Heuristics related to the device in normal operation so that network analysis can be performed*
Secure Device Capabilities – Baseline INDUSTRY ACCEPTED PROTOCOLS ARE USED FOR COMMUNICATIONS	*Device must support industry standard protocols internally and for data transmission egress* An Endpoint that communicates via IEEE 802 networking must support [RFC7030], Section 3 on TLS Layer, for certificate management of secure transport. Endpoints must measure secure boot: Secure boot is a 'security mechanism' and measured boot is the monitoring required Endpoints using IEEE 802.3 (wired Ethernet) must support [IEEE 802.1x] using the EAP-TLS [RFC5216] EAP method. Endpoints that have IEEE 802.11 transceivers MUST make use of [IEEE802.11] security in conjunction with [IEEE802.1X] (WPA Enterprise) to exchange [IEEE802.1AR] certificates
Secure Device Capabilities – Baseline DATA VALIDATION	
Secure Device Capabilities – Baseline EVENT LOGGING	Device must be able to log event and provide secure access to such logs to authorized users- lifecycle management

CATEGORY	MAPS TO
Secure Device Capabilities – Baseline CRYPTOGRAPHY	a. Cryptography: The endpoint MUST support the SHA-256 hash algorithm b. The endpoint must support for Elliptic Curve Cryptography (ECC) described in [RFC6090] and [IEEE802.1AR] for use as LDevIDs c. An Endpoint must support either 2048-bit RSA certificates or ECC certificates as described in [RFC6090] and [IEEE802.1AR] for IDevIDs d. TLS Cipher Suite Support: Endpoints must minimally support the TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8 cipher suite which is detailed within [RFC 7251] for EAP-TLS. This cipher suite will be used for the authentication operations used for both network layer and application layer authentication processes. RNG: An Endpoint must provide random number generation either through hardware or as compliant with FIPS 140-2 Sections 4.7.1 and 4.9.2.
Secure Device Capabilities – Baseline PATCHABILITY	*Device and manufacturer support secure SW/FW/HW updates throughout device lifecycle* a. Endpoints must have the ability to securely receive and apply a software and/or firmware update b. All updates must be signed by the manufacturer, and Endpoints must validate signatures prior to applying any updates c. Endpoints that implement via IEEE 802 networking must support installation of at least one local certificate (LDevIDs) and associated private keying material
Secure Device Capabilities – Baseline REPROVISIONING	*Device must support secure, authorized access control for remote and physical connection to device*
Product Lifecycle Management VULNERABILITY SUBMISSION AND HANDLING PROCESS	*Manufacturer must provide any known device vulnerabilities and a plan or process to mitigate such vulnerabilities* *Endpoint manufacturers must have an active product incident response team (PSIRT), with documented processes and service level agreements, that customers and others can easily locate and call to report vulnerabilities.*
Product Lifecycle Management EOL/EOS UPDATES AND DISCLOSURE	*Manufacturer should provide any EoL and end of support or EoS announcements in a timely manner to device owners. In addition manufactures should provide any expected vulnerabilities expected to E-o-Support (recommendations for mitigation)*
Product Lifecycle Management DEVICE INTENT DOCUMENTATION	
Secure Capabilities – Phase In Over Time DEVICE INTENT SIGNALING	Manufacturer must provide a file server that distributes Manufacturer Usage Description (MUD) files in accordance with RFC 8520 a. When a device certificate is present, the MUD-URL must be included in the client certificate used for a client authenticated 802.1X exchange. If an 802.1X service is not discovered by the client it must also present an unsecured statement of the MUD-URL via LLDP or DHCP b. Endpoints must only run applications or services whose TCP or UDP ports are described in the MUD profile
Secure Capabilities – Phase In Over Time DEVICE NETWORK ONBOARDING	Device must support MUD URLs to provide the network with information to microsegment/set ACL's. In addition the device should support an automated onboarding capability such as BRSKI

CATEGORY	MAPS TO
Additional IoT Device Security Capabilities and Practices SECURE DEVELOPMENT LIFECYCLE	Vendors must have a written SDL process in place that includes the following elements at a minimum: ▪ Training for software developers which includes secure coding techniques and requirements standard C libraries. ▪ Threat modeling that includes a summary report of findings and a diagram. ▪ Software security testing thru either dynamic or static analysis tools and a report that demonstrates testing was completed and output of testing. A way to document and track third party and open source components used in product. A summary of the vendor's specific SDLC process must be available on their public facing webserver.
Additional IoT Device Security Capabilities and Practices HARDWARE ROOTED SECURITY	Secure Storage: The Endpoint must contain its own certificate. The Endpoint must also contain the root certificate for the IDevID, Software Image Signing and Onboarding Services (MASA Root). Total of 4 certificates. Endpoints must store private keying material and certificates in tamperproof storage
Additional IoT Device Security Capabilities and Practices TIME DISTRIBUTION	a. A trusted time source is necessary for the process of certificate validation and reliable system event logging and correlation. Endpoints MUST use either Simple Network Time Protocol (NTP) version 4 [RFC4330] or time provided by a trusted and authenticated server as described in Section 5.5. b. Endpoints must periodically write the current time to non-volatile storage, and use that as a base prior to being configured with accurate time. The purpose of doing so is simply to prevent attackers from using expired certificate to gain unauthorized access to an Endpoint.
Additional IoT Device Security Capabilities and Practices SYSTEM RESILIENCY	Device must be able to function post security attack (based on no damage. May require SW/FW reinstatement or update)
Additional IoT Device Security Capabilities and Practices SECURE TOOLCHAINS	
Additional IoT Device Security Capabilities and Practices SOFTWARE TRANSPARENCY AND BILL OF MATERIALS	Device must be able to store data and provide access to security breaches during the lifecycle
Additional IoT Device Security Capabilities and Practices LEAST FUNCTIONALITY	*Device should provide mitigation options including device shut-down in the event of a security attack/ breach* a. Network elements must support limited network access for endpoints that do not support 802.1X b. Upon detecting a threat, a Network must isolate infected devices based on local policy and report the action to the network administrator.
Additional IoT Device Security Capabilities and Practices PHYSICAL ACCESS CONTROL	*Device should be able to block un-authorized physical access. For direct connection to a device there must be a secure/authorization process* Endpoints must protect personally identifiable information from disclosure and modification. The actual implementation will depend on the nature of the endpoint and associated service, but an example would be to encrypt information on board the device such that only authorized users may access it.
Additional IoT Device Security Capabilities and Practices BEST CURRENT PRACTICES	

11 : Annex G: Mapping to IoXT Pledge

CATEGORY	MAPS TO
Secure Device Capabilities – Baseline DEVICE IDENTIFIERS	
Secure Device Capabilities – Baseline SECURED ACCESS	[Security Pledge] 1. No universal passwords The product shall not have a universal password; unique security credentials will be required for operation. Products shall either have a unique password or require the user to enter a new password immediately upon first use. [Security Pledge] 2. Secured Interfaces All product interfaces shall be appropriately secured by the manufacturer. In all cases, any external communication interfaces shall be secured. For products in which local attacks are a concern, internal chip-to-chip interfaces may be secured.
Secure Device Capabilities – Baseline DATA IN TRANSIT IS PROTECTED	[Security Pledge] 2. Secured Interfaces In all cases, any external communication interfaces shall be secured. All sensitive interfaces shall be encrypted and authenticated.
Secure Device Capabilities – Baseline DATA AT REST IS PROTECTED	
Secure Device Capabilities – Baseline INDUSTRY ACCEPTED PROTOCOLS ARE USED FOR COMMUNICATIONS	[Security Pledge] 3. Proven cryptography Specifically, suitable cryptographic security techniques and algorithms that are well developed, proven, reviewed and standardized and should be applied wherever possible in place of proprietary developed algorithms, which haven't been subjected to the same level of scrutiny and review.
Secure Device Capabilities – Baseline DATA VALIDATION	[Security Pledge] 5. Signed software updates The product shall only support signed software updates. While it is critical that all products be updatable, it is just as critical that these update images be secured. A manufacturer must cryptographically sign update images to prevent tampering during deployment. The product must not use unsigned updates, as they could be fraudulent. [Security Pledge] 2. Secured Interfaces All sensitive interfaces shall be encrypted and authenticated.
Secure Device Capabilities – Baseline EVENT LOGGING	
Secure Device Capabilities – Baseline CRYPTOGRAPHY	[Security Pledge] 3. Proven cryptography Product security shall use strong, proven, updatable cryptography using open, peer-reviewed methods and algorithms ioXt Security Pledge participants agree their product's security shall use proven and standardized cryptography. Specifically, suitable cryptographic security techniques and algorithms that are well developed, proven, reviewed and standardized and should be applied wherever possible in place of proprietary developed algorithms, which haven't been subjected to the same level of scrutiny and review.

CATEGORY	MAPS TO
Secure Device Capabilities – Baseline PATCHABILITY	[Security Pledge] 6. Automatically applied updates The manufacturer will act quickly to apply timely security updates. Whenever a security vulnerability is detected, the manufacturer will automatically apply a patch to the product. No user intervention will be required.
Secure Device Capabilities – Baseline REPROVISIONING	
Product Lifecycle Management VULNERABILITY SUBMISSION AND HANDLING PROCESS	[Security Pledge] 7. Vulnerability reporting program The manufacturer shall implement a vulnerability reporting program, which will be addressed in a timely manner.
Product Lifecycle Management EOL/EOS UPDATES AND DISCLOSURE	[Security Pledge] 8. Security Expiration Date The manufacturer shall be transparent about the period of time that security updates will be provided. Like a manufacturer's product warranty, there shall be transparency around the support period of security updates.
Product Lifecycle Management DEVICE INTENT DOCUMENTATION	
Secure Capabilities – Phase In Over Time DEVICE INTENT SIGNALING	
Secure Capabilities – Phase In Over Time DEVICE NETWORK ONBOARDING	[Security Pledge] 2. Secured Interfaces In all cases, any external communication interfaces shall be secured. For products in which local attacks are a concern, internal chip-to-chip interfaces may be secured. Further, memory interface may also be secured through secure boot or other memory integrity checks. All sensitive interfaces shall be encrypted and authenticated.
Additional IoT Device Security Capabilities and Practices SECURE DEVELOPMENT LIFECYCLE	
Additional IoT Device Security Capabilities and Practices HARDWARE ROOTED SECURITY	[Security Pledge] 2. Secured Interface For products in which local attacks are a concern, internal chip-to-chip interfaces may be secured. Further, memory interface may also be secured through secure boot or other memory integrity checks.
Additional IoT Device Security Capabilities and Practices TIME DISTRIBUTION	
Additional IoT Device Security Capabilities and Practices SYSTEM RESILIENCY	

CATEGORY	MAPS TO
Additional IoT Device Security Capabilities and Practices SECURE TOOLCHAINS	
Additional IoT Device Security Capabilities and Practices SOFTWARE TRANSPARENCY AND BILL OF MATERIALS	[Security Pledge] 8. Security expiration date The manufacturer shall be transparent about the period of time that security updates will be provided. Like a manufacturer's product warranty, there shall be transparency around the support period of security updates.
Additional IoT Device Security Capabilities and Practices LEAST FUNCTIONALITY	[Security Pledge] 4. Security by default Product security shall be appropriately enabled by default by the manufacturer. This principle guarantees that products are appropriately secured at the time of purchase.
Additional IoT Device Security Capabilities and Practices PHYSICAL ACCESS CONTROL	[Security Pledge] 2. Secured Interface For products in which local attacks are a concern, internal chip-to-chip interfaces may be secured. Further, memory interface may also be secured through secure boot or other memory integrity checks.
Additional IoT Device Security Capabilities and Practices BEST CURRENT PRACTICES	

12 : Annex H: Mapping to Open Connectivity Foundation Specifications

The Open Connectivity Foundation (OCF) provides the following mapping of its secure interoperability specification, as of the publication date of this document, to the IoT security capabilities set forth in the above document. OCF continues to revise and expand its specification and associated conformance testing and certification program. To ensure access to the most accurate and up-to-date information on the OCF specification and testing and certification program, please visit https://openconnectivity.org.

CATEGORY	MAPS TO
Secure Device Capabilities – Baseline DEVICE IDENTIFIERS	**[OCF Security Specification ISO/IEC 30118-2:2018] Clause 7.1.1:** Device IDs shall be unique within the scope of operation of the corresponding OCF Security Domain, and should be universally unique.
Secure Device Capabilities – Baseline SECURED ACCESS	**[OCF Security Specification ISO/IEC 30118-2:2018] Clauses: 5.3.3:** Prior to operational state, device must be onboarded and configured with either symmetric or asymmetric credentials based on certificates or shared keys. Once operational devices implement role-based and/or subject based access control for each resource they present to the network. **[OCF Security Specification ISO/IEC 30118-2:2018] Clause 12:** Access control is enforced over all Resources. [**OCF Security Specification ISO/IEC 30118-2:2018] Clause 13.2:** Stored Credentials are used to mutually authenticate servers and clients Physical interface authentication and UI authentication are out of scope for OCF.
Secure Device Capabilities – Baseline DATA IN TRANSIT IS PROTECTED	**[OCF Security Specification ISO/IEC 30118-2:2018] Clause 11.1:** Data in transit devices must support TLS/DTLS version 1.2 or greater for all unicast sessions. **[OCF Security Specification ISO/IEC 30118-2:2018] Cause 11.2:** Cipher Suites: All cipher suites allowed in these specifications are heavily reviewed and IETF approved or greater.
Secure Device Capabilities – Baseline DATA AT REST IS PROTECTED	**[OCF Security Specification] Clause 14.2.2:** Secure storage for credentials is strongly recommended.
Secure Device Capabilities – Baseline INDUSTRY ACCEPTED PROTOCOLS ARE USED FOR COMMUNICATIONS	**[OCF Security Specification ISO/IEC 30118-2:2018] Clause 5 Figure 3:** Shows transport, session and application layer standards. **[OCF Security Specification ISO/IEC 30118-2:2018] Clause 11. 1:** Devices must support CoAP, and CoAP over DTLS. **[OCF Security Specification ISO/IEC 30118-2:2018] Clause 11.2:** Cipher Suites: All heavily reviewed and IETF approved or greater.
Secure Device Capabilities – Baseline DATA VALIDATION	**[OCF Core Technology Specification ISO/IEC 30118-1:2018].** Data model enforcement of encoding, type and length. Data model enforcement occurs on data inbound and outbound to the system. **Certification includes schema validation.**

CATEGORY	MAPS TO
Secure Device Capabilities – Baseline EVENT LOGGING	**[OCF Security Specification] Clause 5.7:** An OCF Platform can generate various kinds of Auditable Events. These Auditable Events can be used for log analysis or for real-time understanding of a system condition.
Secure Device Capabilities – Baseline CRYPTOGRAPHY	**[OCF Security Specification ISO/IEC 30118-2:2018] Clause 11.2:** This clause lists the cipher suites allowed during ownership transfer and normal operation. All cipher suites are recognized IETF RFCs and most are IANA supported ciphers. Strong, proven, updateable cryptography using open, peer-reviewed methods and algorithms. NIST approved algorithms for all cryptographic operations.
Secure Device Capabilities – Baseline PATCHABILITY	**[OCF Vendor Attestation Document]:** Certification Applicant agrees to respond to, address, and patch software vulnerabilities as prescribed by the OCF Security Incident Response Plan. **[OCF Security Specification] Clause 14.5.3:** Process where device validates the software version against a trusted source. **[OCF Security Specification] Clause 14.5.4:** A client with the correct authorization can initiate a software update process.
Secure Device Capabilities – Baseline REPROVISIONING	**[OCF Security Specification ISO/IEC 30118-2:2018] Clause 8.5:** Defines how resources on the device are returned to the manufacturer's default values.
Product Lifecycle Management VULNERABILITY SUBMISSION AND HANDLING PROCESS	**[OCF] Security Working Group Incident Response Plan:** document addresses reporting (web page dedicated to reporting of issues), mitigation, timeframes, communication, emergency/critical fixes, and software deployment.
Product Lifecycle Management EOL/EOS UPDATES AND DISCLOSURE	**[OCF] Updatable Certified Product List:** Website. https://openconnectivity.org/certified-products manufacturers should notify OCF that device is EoL.
Product Lifecycle Management DEVICE INTENT DOCUMENTATION	**[OCF Security Specification] Clause 9.4.2.2.3 End Entity Certificate Profile:** The MUD file pointed to by the URI included in the X.509 certificate includes the following properties referenced in RFC 8520: **[RFC 8520] Section 3.7 system info (https://tools.ietf.org/html/rfc8520#section-3.7):** This is a textual UTF-8 description of the Thing to be connected. The intent is for administrators to be able to see a brief displayable description of the Thing. It SHOULD NOT exceed 60 characters worth of display space. **[RFC 8520] Section 4.3 documentation (https://tools.ietf.org/html/rfc8520#section-4.3):** This URI consists of a URL that points to documentation relating to the device and the MUD file.
Secure Capabilities – Phase In Over Time DEVICE INTENT SIGNALING	**[OCF Security Specification] Clause 9.4.2.2.3 End Entity Certificate Profile:** This section details the manner in which devices can signal intent and capabilities beyond those currently in use for security profiles. MUD URI's can be encoded here, as can attestations about meeting differing hardening requirements, certificate trust chains, and more.
Secure Capabilities – Phase In Over Time DEVICE NETWORK ONBOARDING	**[OCF WiFi Easy Setup Specification]:** Includes the WiFi Easy Setup Resources, and the other transport-level onboarding (e.g. Bluetooth) are defined in other specification documents for OCF. **[OCF Security Specification ISO/IEC 30118-2:2018] Clause 5.2 Onboarding Overview:** For non-transport onboarding, the process is specified in great detail as far as establishment of trust, authentication, verification, authorization, local credential issuance, etc.

CATEGORY	MAPS TO
Additional IoT Device Security Capabilities and Practices SECURE DEVELOPMENT LIFECYCLE	**[OCF Security Specification] Clause 14.2.2.4: Additional Security Guidelines and Best Practices:** Address Software and Secure Development Lifecycle, but OCF is not an application level specification, rather it is a Session-level specification so there will always be additional software added to the foundation OCF provides.
Additional IoT Device Security Capabilities and Practices HARDWARE ROOTED SECURITY	**[OCF Security Specification] Clause 14.8.3.4: Black Security Profile:** Requires the manufacturer to install a certificate which chains to the OCF root certificate (which is in each onboarding tool's trust store) to validate the hardware has been OCF Certified by an authorized test lab, that it chains to that manufacturer's intermediate root, and that it shares a trust relationship bound to the hardware and secure credential store of the device. **[OCF Security Specification] Clause 14.2.2.2:** Hardware Secure storage is recommended for symmetric and asymmetric keys, access credentials and personal private data. **[OCF Security Specification] Clause 14.2.7:** Defines levels of Hardware Tamper Protection for cryptographic module.
Additional IoT Device Security Capabilities and Practices TIME DISTRIBUTION	**[OCF Security Specification] Clause 14.2.5:** Secure time source can be external as long as it is signed by a trusted source and the signature validation in the local device is a trusted process (e.g. backed by secure boot).
Additional IoT Device Security Capabilities and Practices SYSTEM RESILIENCY	**[OCF Security Specification] Clause 5.1:** Shows the layers of connectivity and access control all remain proximally functional by default. **[OCF Security Specification] Clause 14.2.2.2:** Secure Virtual Resources (SVRs) are stored in non-volatile storage. Certification requires that all devices maintain proximal control in the case of a wide area network outage.
Additional IoT Device Security Capabilities and Practices SECURE TOOLCHAINS	**[OCF Security Specification] Clause 14.2.2.4-13: Security Hardening Guidelines/ Execution Environment Security:** It is recommended that at least one static and dynamic analysis tool be applied to any proposed major production release of the software before its release, and any vulnerabilities resolved.
Additional IoT Device Security Capabilities and Practices SOFTWARE TRANSPARENCY AND BILL OF MATERIALS	IoTivity is an open source implementation for OCF and lists all software dependencies. https://iotivity.org/
Additional IoT Device Security Capabilities and Practices LEAST FUNCTIONALITY	**[OCF Security Specification ISO/IEC 30118-2:2018] Clause 12: Access Control:** Employs a deny-all, permit-by-exception policy to allow access to Resources (data and actuators) for Read/Write/Create/Delete/Notify. Access control can be updated dynamically at the location of deployment to limit access (to a role, Device, or implementation).
Additional IoT Device Security Capabilities and Practices PHYSICAL ACCESS CONTROL	**[OCF Security Specification] Clause 14.2.7:** Defines levels of Hardware Tamper Protection for cryptographic module. **[OCF Security Specification] Clause 14.2.2.4:** Additional Security Guidelines and Best Practice
Additional IoT Device Security Capabilities and Practices BEST CURRENT PRACTICES	[OCF Security Specification] Clause 14.2.2.4: Additional Security Guidelines and Best Practices: Discuss non-certifiable/non-testable behaviors that are desirable in software development, hardware development, deployment, testing, and hardening areas.

13 : Annex I: Mapping to UL MCV 1376 – Security Capabilities Verified

CATEGORY	MAPS TO
Secure Device Capabilities – Baseline DEVICE IDENTIFIERS	None.
Secure Device Capabilities – Baseline SECURED ACCESS	**2.1 No default credentials or secret keys** ***System defaults such as password and/or cryptographic keys must be changed on initial setup*** *Base requirement: L1–L5* Ideally, system defaults should be avoided—but realistically that's not always possible. It may be necessary for something to be set to a default value to allow for the "boot-strapping" of the system for the first time. However, the risk of using the default should be clearly outlined to the people operating that system for that first time, and this requirement outlines the need to force them to make a change from this default as part of the overall setup. It is accepted that this can cause problems—forcing people to think up a new password during setup, for example, can lead to a less than secure value being set (or a value that the user will forget the second they walk away). Where system defaults are automatically changed as part of the personalization/manufacturing process, these values must be set such that they are unique per device and statistically uncorrelated between devices (i.e. assigned randomly). Any such "pre-changed" values must also be set in compliance with the password and cryptographic policies of the vendor. Ultimately, defaults for passwords and other such items should only be implemented for values that are absolutely required to be present for normal operation, but which must be changed by the user before operation. This also covers any debugging/backdoor accounts that may be used during development—such values must never be left in a production system. Default and common values for certificates and public keys may be implemented to ease the remote management of systems. For example, it is common to have a globally fixed public key value across devices to authenticate software updates. However, such defaults must be clearly justified as to why they cannot be uniquely assigned per device. **2.2 Protect sensitive data** ***Sensitive data must be protected in transit and at rest*** *Base requirement: L1–L5* Bad-actors will often attempt to recover sensitive data, such as passwords, secret cryptographic keys, and customer data, as the start of an attack on a system. This data may be easily accessed if it is not protected, and electronic protection must always involve strong cryptography and key management to ensure that it is providing the controls at a sufficient level. Therefore any data that is communicated across connections that are not physically direct (such as a direct USB or serial connection) must be protected against disclosure through cryptographic means. *Requirement enhancement 1: L3–L5* Additionally for level 3 and up, storage of such sensitive data must also be protected as customers are likely to re-use passwords across different devices, or even re-purpose online passwords for home use. This includes ensuring that such data is not easily accessible with internal access to the device (e.g., through monitoring an internal serial bus). It is understood that sometimes such data must be displayed for business and user interface reasons (e.g., to display and receive a user password as it is entered), but business justification for each exposure must be provided. Passwords must never be stored in plain text, but instead always in hashed form. When possible the hashing process should include the usage of a salt, where the salt is defined as an unique, randomly generated string that is added to each password before hashing. Weak or broken hashing algorithms, such as MD5, must not be used.

CATEGORY	MAPS TO
	Furthermore, industry standard cryptographic algorithms, outlined in **Chap. 6**, must be used to protect sensitive data. Development of proprietary, or bespoke, algorithms or protections actually weakens systems as such algorithms will not have undergone the many years of academic review and attack that is performed on those industry standard methods. Therefore, protections can only be assumed when such standard algorithms are used. *Requirement enhancement 2: L4, L5* For implementation that target security level 4 and up, the hashing process must incorporate a key stretching algorithms such as scrypt or PBKDF2 to reduce the susceptibility against brute-force attacks. **2.3 Passphrase complexity enforcement** ***When passphrases are used to authorize use of services, they must fulfill minimum strength criteria*** *Base requirement: L2–L5* Passphrases are often required and implemented to provide authentication of users. If not set to a value that is sufficiently secure, they can be easily guessed or brute-forced to bypass this authentication, allowing a bad actor to gain access to the services the passphrases are supposed to protect. Many attacks on devices are based on exploiting insecure, or default, password values. The strength of a passphrase typically depends on two key factors: The first factor is the set of characters that passphrase characters are chosen from, known as the alphabet. The second factor is the length of the passphrase in characters. Strengths of passphrases are typically given in equivalent bit lengths, i.e., the binary logarithm of the number of possible combinations. This requirement deals with the complexity of such passwords. Note that the requirement does only apply to scenarios in which they are technically feasible. For example, a numeric number verification for Bluetooth pairing would be out of scope in regards to this requirement, because in principle the underlying system prevents using sufficiently secure passcodes. Similar situations arise where a passphrase may only use a limited alphabet because the HMI does not allow other characters (e.g., a device with keypad that only has digits 0-9). We differentiate in this requirement passphrases that are chosen by a user (i.e., a passphrase that the user can change themselves) against those that are chosen by a device or machine entirely at random (e.g., an API key that is chosen by software and cannot be changed by a user). For the latter, the plausibilization routines used for human-choice passwords must not be used, since that would counterintuitively decrease the strength of passphrases. Users should always be allowed to at least use the 26 special characters that correspond to ASCII codepoints 0x21 - 0x2f (!”#$%&’()*+,-./), 0x3a - 0x3f (:;<=>?), and 0x5b - 0x5f ([\]^_) if they so choose. Furthermore, the maximum length of a password shall not be restricted below 127 characters of length, meaning that any system shall be able to accept a password up to 127 characters (but may of course support longer passphrases). For the base requirement, these are the minimum criteria regarding passphrase complexity: ▪ User-chosen passphrases: length at least 10 characters, at least one uppercase, and at least one lowercase character. Example of valid passphrases: “Achievement”, waterFaLL5”. Example of invalid passphrases: “fooBar123” (too short), “achievement” (no uppercase character), “ACHIEVEMENT” (no lowercase character). ▪ Machine-chosen passphrases: alphabet at least [A-Za-z], length at least 10 characters (approximately 57 bit of security) *Requirement enhancement 1: L4, L5* ▪ For level 4 and up, those rules become stricter: ▪ User-chosen passphrases: length at least 12 characters, at least one uppercase, at least one lowercase character, and at least one digit. Examples of valid passphrases: “Y3ll0whamm3r”, “M1croorgan1sm”. Example of invalid passphrases: “foobarF0BAR” (too short), “ucroorgan1sm” (no uppercase character), “UCROOGRAN1SM” (no lowercase character), “FOOBARfoobar” (no digit). ▪ Machine-chosen passphrases: alphabet at least [A-Za-z0-9], length at least 12 characters (approximately 71 bit of security)

CATEGORY	MAPS TO
	3.1 Disable debug interfaces ***Debug interfaces must be disabled or protected against misuse*** *Base requirement: L3–L5* Often devices will come with some interfaces that are either specifically designed, or can be used, for "debugging" purposes. For example, local JTAG ports can often be used to extract software from devices and start the reverse engineering process which allows for determination of vulnerabilities within the device. Another example are serial connections (e.g., via RS232) which may output sensitive information or provide direct access to the system. Such "debug" interfaces must be disabled in production devices. If it is not possible to do so, mitigations must be implemented to prevent extraction of sensitive data or exploitation of the device. Often microcontrollers provide a feature called CRP which prevents the internal flash read out via JTAG. Devices which support this feature must have it enabled. To fulfill level 3, at least serial connections must be deactivated or protected. *Requirement enhancement 1: L4, L5* Additionally for level 4 and up, JTAG connections must be deactivated or protected. **4.1 Sensitive services require authentication** ***Sensitive services must require authentication and ensure the confidentiality and integrity of data*** *Base requirement: L1–L5* Sensitive services within a device are considered to be services which allow for the allocation or changing of security settings, or which allow for access to customer personal information (such as authentication data, email addresses, etc.). Such access is inherently security sensitive, and therefore requires authentication to be performed to ensure that any changes are being correctly performed by the customer, and are not being accessed or altered by a bad-actor. This includes ensuring that access, once authenticated, ensures the integrity of data as it is passed into the device, as well as ensuring confidentially of any customer data during transport It is considered likely that many systems will rely on standard protocols such as TLS to provide these features, and testing will include validating that such protocols are correctly configured and used, along with ensuring that weak modes of operation—such as insecure cipher suites—are disabled by default. *Requirement enhancement 1: L4, L5* For devices that target security level 4 and up, sensitive services must be protected against brute force attacks by providing limits for authentication attempts.
Secure Device Capabilities – Baseline DATA IN TRANSIT IS PROTECTED	**2.2 Protect sensitive data** ***Sensitive data must be protected in transit and at rest*** *Base requirement: L1–L5* Bad-actors will often attempt to recover sensitive data, such as passwords, secret cryptographic keys, and customer data, as the start of an attack on a system. This data may be easily accessed if it is not protected, and electronic protection must always involve strong cryptography and key management to ensure that it is providing the controls at a sufficient level. Therefore any data that is communicated across connections that are not physically direct (such as a direct USB or serial connection) must be protected against disclosure through cryptographic means. *Requirement enhancement 1: L3–L5* Additionally for level 3 and up, storage of such sensitive data must also be protected as customers are likely to re-use passwords across different devices, or even re-purpose online passwords for home use. This includes ensuring that such data is not easily accessible with internal access to the device (e.g., through monitoring an internal serial bus). It is understood that sometimes such data must be displayed for business and user interface reasons (e.g., to display and receive a user password as it is entered), but business justification for each exposure must be provided. Passwords must never be stored in plain text, but instead always in hashed form. When possible the hashing process should include the usage of a salt, where the salt is defined as an unique, randomly generated string that is added to each password before hashing. Weak or broken hashing algorithms, such as MD5, must not be used.

CATEGORY	MAPS TO
	Furthermore, industry standard cryptographic algorithms, outlined in **Chap. 6**, must be used to protect sensitive data. Development of proprietary, or bespoke, algorithms or protections actually weakens systems as such algorithms will not have undergone the many years of academic review and attack that is performed on those industry standard methods. Therefore, protections can only be assumed when such standard algorithms are used. *Requirement enhancement 2: L4, L5* For implementation that target security level 4 and up, the hashing process must incorporate a key stretching algorithms such as scrypt or PBKDF2 to reduce the susceptibility against brute-force attacks. **5.1 Cryptographically Secure Data Transmission** ***Communication channels need to be protected via cryptographic means to achieve various security properties*** *Base requirement: L1–L5* Any communication channel through which unintended actions can be triggered must be secured in a way that achieves secure communication even when the medium used for transmission cannot be considered secure. For instance, communication over the Internet could potentially be read and modified by anyone on the routing path. An end-to-end security implementation would ensure that the communication still retains important security properties, namely: ▪ Confidentiality of data: An eavesdropper on the connection is unable to make sense of the transmitted information ▪ Integrity of data: It is possible to determine with exceeding likelihood if received data was modified in transit ▪ Peer validation: The respective peer on the other end of the connection can be verified to be the correct party with whom communication is intended ▪ Downgrade protection: The protocol, if it supports multiple versions, must always use a version both peers agree on and may not be artificially downgraded by an adversary ▪ Replay protection: Data that has previously been recorded by an adversary and that is repeated by that adversary is detected as a duplicate and properly rejected Typically, this is achieved by using TLS as the foundational transport protocol, which, in a correct configuration, can achieve all of these security protocols. Note, however, that even a TLS configuration can be susceptible to attacks on these security goals; most notably if poor choices in the protocol parameterization are used (e.g., weak cipher suites), specific security mechanisms are disabled (e.g., peer validation). Replay protection may be deliberately sacrificed in specific scenarios as well. One example of this would be the use of the 0RTT feature of TLSv1.3. This is permissible if and only if the concerned software has other means of ensuring the replay of messages does not impact the overall security of the system. Specific resource constraints lead to a situation in which deeply embedded devices may not have the resources to fulfill a full TLS handshake; they still need to make sure that the desired security properties are met. *Requirement enhancement 1: L2–L5* For devices that target security level 2 and up, the implementation must either follow an industry-standard security protocol (such as TLS) or a proof of the security properties must be provided that has been vetted by experts in the field. Note that this is typically a task that is exceedingly difficult to achieve because of the required expertise in the field of theoretical cryptography and cryptanalysis. *Requirement enhancement 2: L3–L5* For devices that target security level 3 and up, custom cryptographic constructions are disallowed and industry-standard protocols must be used either way. Furthermore, for these devices, it is required that all secured communication that falls under this clause also achieves Perfect Forward Secrecy (PFS).

CATEGORY	MAPS TO
Secure Device Capabilities – Baseline DATA AT REST IS PROTECTED	**2.2 Protect sensitive data** ***Sensitive data must be protected in transit and at rest*** *Base requirement: L1–L5* Bad-actors will often attempt to recover sensitive data, such as passwords, secret cryptographic keys, and customer data, as the start of an attack on a system. This data may be easily accessed if it is not protected, and electronic protection must always involve strong cryptography and key management to ensure that it is providing the controls at a sufficient level. Therefore any data that is communicated across connections that are not physically direct (such as a direct USB or serial connection) must be protected against disclosure through cryptographic means. *Requirement enhancement 1: L3–L5* Additionally for level 3 and up, storage of such sensitive data must also be protected as customers are likely to re-use passwords across different devices, or even re-purpose online passwords for home use. This includes ensuring that such data is not easily accessible with internal access to the device (e.g., through monitoring an internal serial bus). It is understood that sometimes such data must be displayed for business and user interface reasons (e.g., to display and receive a user password as it is entered), but business justification for each exposure must be provided. Passwords must never be stored in plain text, but instead always in hashed form. When possible the hashing process should include the usage of a salt, where the salt is defined as an unique, randomly generated string that is added to each password before hashing. Weak or broken hashing algorithms, such as MD5, must not be used. Furthermore, industry standard cryptographic algorithms, outlined in , must be used to protect sensitive data. Development of proprietary, or bespoke, algorithms or protections actually weakens systems as such algorithms will not have undergone the many years of academic review and attack that is performed on those industry standard methods. Therefore, protections can only be assumed when such standard algorithms are used. *Requirement enhancement 2: L4, L5* For implementation that target security level 4 and up, the hashing process must incorporate a key stretching algorithms such as scrypt or PBKDF2 to reduce the susceptibility against brute-force attacks.
Secure Device Capabilities – Baseline INDUSTRY ACCEPTED PROTOCOLS ARE USED FOR COMMUNICATIONS	**5.1 Cryptographically Secure Data Transmission** ***Communication channels need to be protected via cryptographic means to achieve various security properties*** *Base requirement: L1–L5* Any communication channel through which unintended actions can be triggered must be secured in a way that achieves secure communication even when the medium used for transmission cannot be considered secure. For instance, communication over the Internet could potentially be read and modified by anyone on the routing path. An end-to-end security implementation would ensure that the communication still retains important security properties, namely: ▪ Confidentiality of data: An eavesdropper on the connection is unable to make sense of the transmitted information ▪ Integrity of data: It is possible to determine with exceeding likelihood if received data was modified in transit ▪ Peer validation: The respective peer on the other end of the connection can be verified to be the correct party with whom communication is intended ▪ Downgrade protection: The protocol, if it supports multiple versions, must always use a version both peers agree on and may not be artificially downgraded by an adversary ▪ Replay protection: Data that has previously been recorded by an adversary and that is repeated by that adversary is detected as a duplicate and properly rejected

CATEGORY	MAPS TO
	Typically, this is achieved by using TLS as the foundational transport protocol, which, in a correct configuration, can achieve all of these security protocols. Note, however, that even a TLS configuration can be susceptible to attacks on these security goals; most notably if poor choices in the protocol parameterization are used (e.g., weak cipher suites), specific security mechanisms are disabled (e.g., peer validation). Replay protection may be deliberately sacrificed in specific scenarios as well. One example of this would be the use of the 0RTT feature of TLSv1.3. This is permissible if and only if the concerned software has other means of ensuring the replay of messages does not impact the overall security of the system. Specific resource constraints lead to a situation in which deeply embedded devices may not have the resources to fulfill a full TLS handshake; they still need to make sure that the desired security properties are met. *Requirement enhancement 1: L2–L5* For devices that target security level 2 and up, the implementation must either follow an industry-standard security protocol (such as TLS) or a proof of the security properties must be provided that has been vetted by experts in the field. Note that this is typically a task that is exceedingly difficult to achieve because of the required expertise in the field of theoretical cryptography and cryptanalysis. *Requirement enhancement 2: L3–L5* For devices that target security level 3 and up, custom cryptographic constructions are disallowed and industry-standard protocols must be used either way. Furthermore, for these devices, it is required that all secured communication that falls under this clause also achieves Perfect Forward Secrecy (PFS).
Secure Device Capabilities – Baseline DATA VALIDATION	**4.4 Input validation and sanitization** ***External inputs must be validated and sanitized before evaluation or execution*** *Base requirement: L4, L5* Functionality that allows for the direct execution of code or commands by the device or system can often be exploited by a malicious party. Such functionality should not be natively supported, and any method which passes user supplied inputs to a system shell or parses and evaluates it directly with its native interpreter must validate and sanitize it beforehand. This not only covers direct inputs such as form fields or file uploads but also any other input data the method receives (HTTP headers, cookies, query strings, SQL queries, formatted payload data such as JSON, XML, CSV, JPEG, etc.). In this context input validation ensures that inputs conform to requirements such as length and data type of the receiving method, whereas input sanitization ensures that inputs conform to requirements of the underlying system to which the inputs are passed. This may include elimination of unwanted characters by means of removing, replacing, encoding or escaping characters. If possible, it is preferred to use the command interpreters or parsers provided functionalities for input validation and sanitization over custom implementations. This requirement covers all interfaces and services which receive and handle device external inputs.
Secure Device Capabilities – Baseline EVENT LOGGING	**3.7 Logs or errors do not expose sensitive data** ***Logging and error messages must not expose sensitive data without authentication*** *Base requirement: L4, L5* It is often necessary for systems to be placed into a “debug” or “logging” mode to facilitate the identification and remediation of problems with the device. However, such data may be used to gain information about the system, or to obtain data that should otherwise remain confidential. Therefore, it is important that any functions that allow for the logging of sensitive data are disabled by default and can only be temporarily enabled after suitable authentication. Once enabled, such logging should not remain active for more than 15 minutes, to ensure that the logging state is not accidentally left active. It is also strongly recommended that any sensitive data that is logged is secured with cryptography (e.g., through encryption using a public key on the device). Any upload or exfiltration of user identifiable data from the customer premises in such logs must be covered under the privacy policy of the system, and require an opt-in from the customer to accept the transfer of this data.

CATEGORY	MAPS TO
	Error messages may also result in the exposure of information—for example, detailing an error with the padding in a cryptographic message can sometimes help attackers determine the values of sensitive information. Therefore, error messages must be carefully designed to not expose details that are too specific about the error state, and instead simply inform the user that an error has occurred. Timing of error messages must also be carefully managed; for example, common compare functions will return an error as quickly as they can, and therefore if used in comparison functions on sensitive data (e.g., passwords) could accidentally expose information about how many characters of the sensitive data are in fact the same. For this reason, non-timing dependent compare functions are recommended for use with sensitive information, and passwords must not be compared directly with stored plaintext (instead comparing against a hashed value, such as that calculated through the scrypt or PBKDF2 function).
Secure Device Capabilities – Baseline CRYPTOGRAPHY	**2.2 Protect sensitive data** ***Sensitive data must be protected in transit and at rest*** *Base requirement: L1–L5* Bad-actors will often attempt to recover sensitive data, such as passwords, secret cryptographic keys, and customer data, as the start of an attack on a system. This data may be easily accessed if it is not protected, and electronic protection must always involve strong cryptography and key management to ensure that it is providing the controls at a sufficient level. Therefore any data that is communicated across connections that are not physically direct (such as a direct USB or serial connection) must be protected against disclosure through cryptographic means. *Requirement enhancement 1: L3–L5* Additionally for level 3 and up, storage of such sensitive data must also be protected as customers are likely to re-use passwords across different devices, or even re-purpose online passwords for home use. This includes ensuring that such data is not easily accessible with internal access to the device (e.g., through monitoring an internal serial bus). It is understood that sometimes such data must be displayed for business and user interface reasons (e.g., to display and receive a user password as it is entered), but business justification for each exposure must be provided. Passwords must never be stored in plain text, but instead always in hashed form. When possible the hashing process should include the usage of a salt, where the salt is defined as an unique, randomly generated string that is added to each password before hashing. Weak or broken hashing algorithms, such as MD5, must not be used. Furthermore, industry standard cryptographic algorithms, outlined in **Chap. 6**, must be used to protect sensitive data. Development of proprietary, or bespoke, algorithms or protections actually weakens systems as such algorithms will not have undergone the many years of academic review and attack that is performed on those industry standard methods. Therefore, protections can only be assumed when such standard algorithms are used. *Requirement enhancement 2: L4, L5* For implementation that target security level 4 and up, the hashing process must incorporate a key stretching algorithms such as scrypt or PBKDF2 to reduce the susceptibility against brute-force attacks.
Secure Device Capabilities – Baseline PATCHABILITY	**1.1 Remote software updates supported** ***Software updates must be supported, using network or wireless interfaces where available*** *Base requirement: L1–L5* No matter how well software is designed or tested, there will always be bugs and vulnerabilities that are missed. This is just a fact of software development and the sheer complexity of any body of code. So, the update of the software must be allowed in any device to ensure that it can be patched when any such bugs are found. It is an additional requirement that the software update must be able to be performed across a wireless or network interface, should the device provide such an interface. This increases the ease of use for the customer, removing disincentives to install updates. Security of the wireless and/or network interfaces is of course also important, and this is covered in later requirements.

CATEGORY	MAPS TO
	This requirement ensures that such updates are possible, minimizing the risk that devices become permanently vulnerable through new attack methods that are discovered after the initial evaluation/shipping of the device. Checking for malicious changes to the software update is not covered under this requirement, and is instead addressed in a later requirement. For devices which fulfill level 1 at least the applications must be updatable. *Requirement enhancement 1: L2–L5* For level 2 and up, additionally the operating system and firmware must be updatable. *Requirement enhancement 2: L4, L5* For level 4 and up, also second stage bootloader components (i.e., those that are software-updatable as e.g., U-Boot) must be updatable. Bootloaders/MLOs that are in hardware are excluded. **1.3 Software Update Authentication** ***Software updates must be cryptographically authenticated, and provide anti-roll back features*** *Base requirement: L1–L5* Although it is important to support software updates to ensure that devices can be patched and maintained in the field, such features can lead to additional vulnerabilities—where a "bad actor" can install their own software into the device to take control of the device or to prevent its normal operation. To prevent this, it's important that any software update is cryptographically authenticated. Often this will be implemented by using a digital signature across the software image, which can be checked by the original software (or bootloader of the device) either prior to installation or at boot-up. Using a digital signature based on a public key algorithm (such as EdDSA, ECDSA or RSA) ensures that the devices themselves don't need the private key that is used to generate the authentication data. It is additionally required that the update implements "anti-rollback" features—such as a "monotonic" version number included in each release (that is a version number that only increases with each version) which is also checked during installation to ensure that a bad actor can't just install a previous version of software; to "reinstate" any otherwise patched vulnerabilities. This anti-rollback functionality may be waived for patches which only offer different functionalities, but do not patch vulnerabilities. For example, the switch back and forth between two different software flavors is allowed, but as soon as an update incorporates a fix for a security vulnerability, going back to the vulnerable version must be disallowed. If the device does not computationally permit the use of public-key cryptography for securing the integrity of the software, and device-unique symmetric software authentication keys are also not feasible from the perspective of organizational overhead, for level 1 devices shared symmetric keys are permissible under the following constraints: ▪ The embedded device must utilize readout protection of the program memory, e.g., using SoC-provided fuse bits which can lock the readout of the software. This means that an attacker will at least have to break this security feature to retrieve the symmetric key. ▪ The software image that is transmitted during an update does not contain the shared symmetric key (e.g., if the bootloader contains said key and the bootloader itself is not part of the software image) or the whole software image is protected by strong encryption. For these purposes, the algorithms listed in **Chap. 6** are considered acceptable. When an AEAD cipher is used, the authentication tag itself can be used instead of a separate MAC computation. ▪ The encryption and/or MAC keys may never be used for any other purpose but software update. *Requirement enhancement 1: L2–L5* Where a symmetric key system—such as an HMAC—is used for update authentication, the secret key in each device must be unique per device. Otherwise once the software of one device is exposed (e.g., through a physical attack on one device), a valid software signature for all other devices of this type can be created. Therefore, public-key cryptography is recommended to avoid the complexities of managing unique symmetric keys across device portfolios.

CATEGORY	MAPS TO
	Requirement enhancement 2: L4, L5 For target security level 4 and up, the software update image must not only be authenticated but encrypted as well. Since device-individual firmware images are not desirable, it is acceptable for this single key to be symmetric and shared across all devices. It may not be exposed in any firmware update, however (i.e., must always be omitted from the image or only contained in the encrypted portion of the image). The algorithms that are considered acceptable for this purpose are listed in **Chap. 6**.
Secure Device Capabilities – Baseline REPROVISIONING	**4.2 Permanent erasure of sensitive data** ***Permanent erasure of sensitive data must be supported*** *Base requirement: L1–L5* Devices must protect sensitive data even during decommissioning (e.g., to prevent the exposure of customer Wi-Fi passwords after disposal or resale), and therefore implement either a "factory reset" which permanently erases all data and configuration from the device, or provide strong protections to the data even given unrestricted physical access to the device. Where the device supports a network interface, it must be possible to "remotely decommission" the device. At all times, a local decommission procedure must always be provided—this may be passive; e.g., erasure of RAM storage after disconnection from power, but where passive mechanisms are implemented they must operate within less than 8 hours and be shown to ensure permanent erasure.
Product Lifecycle Management VULNERABILITY SUBMISSION AND HANDLING PROCESS	**6.4 Vulnerability management program** ***A vulnerability management and disclosure program must be maintained*** *Base requirement: L1–L5* It can be expected that new issues will become apparent in systems after evaluation and shipping to the customer. Therefore, it is necessary for system vendors to ensure that they have a vulnerability management and disclosure program to maintain the security of their products once shipped. This program must include processes for: ▪ Monitoring for new vulnerabilities in all code that it contained in the software composition list ▪ Testing if vulnerabilities affect the vendor systems, and how they can be mitigated if the system is affected ▪ The creation and testing of a patch for the vulnerability if required ▪ Informing customers of an already published vulnerability, and any mitigating steps they can take whilst a patch is being created. As long as the vulnerability has not become public knowledge yet, it is acceptable to delay informing customers until after the patch has been created. Additionally, contact information and details about the vulnerability disclosure process for external security researchers should be published on a publicly available website
Product Lifecycle Management EOL/EOS UPDATES AND DISCLOSURE	**6.3 End-of-life policy** ***Information on the minimum support period must be available to end users*** *Base requirement: L1–L5* End users shall be able to obtain information on the minimum support period where the manufacturer of the product shall continue to provide software updates to the product. This period is expected to be appropriate to the device, where e.g. devices with a long product lifecycle will continue to receive updates for several years after purchase.
Product Lifecycle Management DEVICE INTENT DOCUMENTATION	None.

CATEGORY	MAPS TO
Secure Capabilities—Phase In Over Time DEVICE INTENT SIGNALING	None.
Secure Capabilities—Phase In Over Time DEVICE NETWORK ONBOARDING	**5.1 Cryptographically Secure Data Transmission** ***Communication channels need to be protected via cryptographic means to achieve various security properties*** *Base requirement: L1–L5* Any communication channel through which unintended actions can be triggered must be secured in a way that achieves secure communication even when the medium used for transmission cannot be considered secure. For instance, communication over the Internet could potentially be read and modified by anyone on the routing path. An end-to-end security implementation would ensure that the communication still retains important security properties, namely: ▪ Confidentiality of data: An eavesdropper on the connection is unable to make sense of the transmitted information ▪ Integrity of data: It is possible to determine with exceeding likelihood if received data was modified in transit ▪ Peer validation: The respective peer on the other end of the connection can be verified to be the correct party with whom communication is intended ▪ Downgrade protection: The protocol, if it supports multiple versions, must always use a version both peers agree on and may not be artificially downgraded by an adversary ▪ Replay protection: Data that has previously been recorded by an adversary and that is repeated by that adversary is detected as a duplicate and properly rejected Typically, this is achieved by using TLS as the foundational transport protocol, which, in a correct configuration, can achieve all of these security protocols. Note, however, that even a TLS configuration can be susceptible to attacks on these security goals; most notably if poor choices in the protocol parameterization are used (e.g., weak cipher suites), specific security mechanisms are disabled (e.g., peer validation). Replay protection may be deliberately sacrificed in specific scenarios as well. One example of this would be the use of the 0RTT feature of TLSv1.3. This is permissible if and only if the concerned software has other means of ensuring the replay of messages does not impact the overall security of the system. Specific resource constraints lead to a situation in which deeply embedded devices may not have the resources to fulfill a full TLS handshake; they still need to make sure that the desired security properties are met. *Requirement enhancement 1: L2–L5* For devices that target security level 2 and up, the implementation must either follow an industry-standard security protocol (such as TLS) or a proof of the security properties must be provided that has been vetted by experts in the field. Note that this is typically a task that is exceedingly difficult to achieve because of the required expertise in the field of theoretical cryptography and cryptanalysis. *Requirement enhancement 2: L3–L5* For devices that target security level 3 and up, custom cryptographic constructions are disallowed and industry-standard protocols must be used either way. Furthermore, for these devices, it is required that all secured communication that falls under this clause also achieves Perfect Forward Secrecy (PFS).
Additional IoT Device Security Capabilities and Practices SECURE DEVELOPMENT LIFECYCLE	**3.5 Software free from known vulnerabilities** ***System software should be free of publicly disclosed vulnerabilities*** *Base requirement: L4, L5* It is increasingly common for systems to be composed of various types and sources of software—from internally developed, to externally developed open source or commercial software. For any externally developed software component, it is possible—and indeed likely—that there are previously disclosed vulnerabilities which have been patched and/or mitigated in further updates to the software. Therefore, it is an essential part of securing software to first identify what externally developed software components exist, and using this list to confirm that these components are up to date and sufficiently mitigate any previously identified vulnerabilities.

CATEGORY	MAPS TO
	It should be noted that—although it is desirable—it is not an absolute requirement that the very latest version is always used if existing vulnerabilities have been mitigated in other ways. **3.6 Software tested for unknown vulnerabilities** ***System software must be tested to check for undisclosed vulnerabilities*** *Base requirement: L5* Although much software may be re-used from other sources, it is unlikely that a device will contain absolutely no internally developed code. In addition, the combination of different software components can open up new threat vectors and potential vulnerabilities. Therefore, it is important that some checking is performed against the software of a device in an attempt to identify such vulnerabilities. The intent of this testing is not to perform an exhaustive penetration test against all features and code of the device, as this would be expensive in terms of both time and direct costs—but to confirm that simple attacks are not possible on the system.
Additional IoT Device Security Capabilities and Practices HARDWARE ROOTED SECURITY	**1.5 Hardware root of trust** ***Device implements a hardware based root of trust for securely storing sensitive data*** *Base requirement: L5* A hardware root of trust (e.g., secure element, TPM) is a dedicated embedded component/memory area which is able to securely store sensitive data such as cryptographic keys. It is the foundation on which all secure operations depend on and a source which can always be trusted by the system and is therefore a crucial component for processes such as the update or boot authentication process, providing a protected environment for encryption and signature verification keys. Having a hardware root of trust increases overall system security as it is exceedingly more difficult to extract or modify its stored data as compared to storing the data in software. To fulfill this requirement the device must store sensitive data, such as private keys (e.g., TLS client certificate keys) in such a hardware root of trust.
Additional IoT Device Security Capabilities and Practices TIME DISTRIBUTION	None.
Additional IoT Device Security Capabilities and Practices SYSTEM RESILIENCY	**4.3 Manual back-up/override for safety critical operations** ***Manual backup/override must be provided for safety related services*** *Base requirement: L2–L5* Safety related services, such as those performed by door locks, are increasingly being automated and enabled through digital systems. This requirement outlines the need of such systems to provide is a safety mechanism that ensures any failure of the device—either through malware, lack of power, or coding flaw—does not result in a safety issue that could lead to risk of life. For example, door locks should provide a manual method for locking and unlocking (such as a "standard" key).
Additional IoT Device Security Capabilities and Practices SECURE TOOLCHAINS	**3.4 Memory and compiler protection** ***Memory and compiler protections must be implemented*** *Base requirement: L4, L5* Modern processing systems and compilers provide multiple methods to assist in the exploitation of any vulnerabilities which may exist in the source code of the device. By correctly enabling and implementing such protections, the security posture of the system can be greatly increased. This requirement does not seek to mandate which protections should be implemented, as this will depend on the specific processing system/operating system/and compiler used—for example, Address Space Layout Randomisation may be implemented in many modern, complex operating systems, but is often not used in smaller Real Time Operating Systems which can have other protection methods. However, it is essential that the vendor demonstrate an understanding of the protections that are available and justify the use (or lack of use) of the protections that they have chosen to implement.

CATEGORY	MAPS TO
Additional IoT Device Security Capabilities and Practices SOFTWARE TRANSPARENCY AND BILL OF MATERIALS	**6.6 Software Bill-of-Materials** ***A Software Composition List must be maintained*** *Base requirement: L4, L5* Any software of sufficiently complexity will contains bugs. It is not possible for any amount of testing to find, and allow for the remediation of, all bugs in any reasonably sized body of code—which is why on-going maintenance of such code is so important. However, it is increasingly common today for the software in a device to be created from various "software components"—open source code, third party libraries, and external binary files. Therefore, in order to maintain code it is not sufficient to simply maintain the code that has been created directly by the product vendor; it is necessary to ensure that all additional "software components" are maintained and updated as well. To achieve this, it is necessary to create and keep up to data a "software composition list" (sometimes called a "software bill of materials") which indicates all of the different software components used in a particular build, as well as their versions. This list must be exhaustive; think of it as an ingredient list for your software, if all of the ingredients are not listed, the recipe will not turn our correctly. In this instance, if not all software is listed, you will not be able to securely maintain your device. Using this software composition list is a prerequisite for the establishment of a vulnerability management program. Using such a program, it is possible to ensure that when there is a new vulnerability found in some third party or open source code that is used in the device, it can be noted, investigated, and patched where necessary.
Additional IoT Device Security Capabilities and Practices LEAST FUNCTIONALITY	**3.2 Systems configured to secure defaults** ***Systems must be configured to secure defaults*** Base requirement: L3–L5 The default configuration of the system must ensure that the device is secure "out of the box". Where deprecated or vulnerable features may be required, or desirable, for a specific market or customer segment, these features must be disabled by default. Examples of such features may be to allow for WPS wireless key negotiation, which is known to be vulnerable but may be desirable in some instances or the usage of Modbus TCP which does not provide authenticity or confidentiality of data but is needed for a particular network environment. Sufficient user guidance should be provided to allow for that user to understand the risks associated with enabling any such features of the device. Devices must also be free from undocumented features that may allow for a "takeover" of the device by someone other than the intended end-users. The system must provide clear documentation of its features, and such "back-door" access or control features must not be implemented or possible within production devices. **3.8 Least Privilege Principle** ***Systems must implement the least privilege principle and utilize hardware-supported features such as memory containment*** *Base requirement: L5* All software of sufficient complexity has vulnerabilities, and "defense in depth" measures must be used to protect against the successful exploitation of any newly discovered flaws. The goal is to have multiple layers of defense so that if one protection mechanism fails in practice, this does not lead to a full system compromise. One good measure is the application of the "least privilege principle" in systems. This means that software is assigned only the execution privilege and access rights that are sufficient and essential for its required operation. Modern processing and operating systems provide many different methods for this to be achieved, and this requirement is not intended to mandate a specific implementation, but instead ensure that the device vendor has considered what access rights are necessary and put in place measures to ensure that additional access is prevented, or at least mitigated. For example, "sandboxing" or virtualized environments may be used, or access between assets and functions may be managed through assigning lower processor and/or operating system privilege levels to all code that does not require full access to the hardware of the device.

CATEGORY	MAPS TO
	Typical means of implementation of this on an embedded system would include that different processes should run as unprivileged users (e.g., the "nobody" user of a Linux system), use of chroot environments, and using file system permissions that disallow access to any data that needs not to be read or written by the respective processes. This requirement would be considered failed if one or more processes were running with root privileges even though they do not require these privileges at runtime. Another failure to meet this requirement would be world-readable (or group-readable) data that is potentially sensitive such as cryptographic keys. **6.7 Physical Interface Documentation** ***All physical interfaces present in the device hardware must be documented and justified*** *Base requirement: L5* The security posture of a system is often described as its "attack surface"—the amount of code that can be interacted with is generally directly related to the potential vulnerabilities a system may have. The more code, the more potential vulnerabilities. However, access to this code is of course also important, and the interfaces of a device are the "front line" of the device security, and by definition attacks on devices generally start with these interfaces. Indeed, any device can be summarized by the totality of its inputs, outputs, and internal processing (where the inputs and outputs are the interfaces). Therefore, it is important for all interfaces of the devices to be clearly understood and justified as to their purpose, as an unnecessary interface may be the one that is used to compromise the system. This list of interfaces must include both physical ports (USB, Serial, Ethernet, etc.) and protocols which are supported over these interfaces. It is recognized that documenting all protocols supported can be quite complex; for example a USB interface may support many different protocols, classes, and types of devices. However, the goal is to ensure that the totality of the interfaces is well understood. **4.6.8 All services documented** ***All services present in the device must be documented and justified*** *Base requirement: L5* For the purposes of this standard a service is considered a super-set of a protocol, in that it actively "listens" for connections across switched or wireless connections. Direct physical interfaces, such as serial or JTAG, are generally considered not to be a "service". As with protocols, listening services are often the first point of attack on a device, and therefore can be the first line of defense to prevent such attacks. Justification of enabled services is vital to understand the security posture of the system, and ensure that sufficient security measures are put in place to protect these interfaces It is understood that additional services may be included in a device as a product differentiator, or to provide value-added services to specific market segments. It is recommended that consideration be given to limiting the functionality of the system "out of the box" and instead providing options for users to enable features where they see a need.
Additional IoT Device Security Capabilities and Practices PHYSICAL ACCESS CONTROL	None.

CATEGORY	MAPS TO
Additional IoT Device Security Capabilities and Practices BEST CURRENT PRACTICES	**5.1 Cryptographically Secure Data Transmission** ***Communication channels need to be protected via cryptographic means to achieve various security properties*** *Base requirement: L1–L5* Any communication channel through which unintended actions can be triggered must be secured in a way that achieves secure communication even when the medium used for transmission cannot be considered secure. For instance, communication over the Internet could potentially be read and modified by anyone on the routing path. An end-to-end security implementation would ensure that the communication still retains important security properties, namely: ▪ Confidentiality of data: An eavesdropper on the connection is unable to make sense of the transmitted information ▪ Integrity of data: It is possible to determine with exceeding likelihood if received data was modified in transit ▪ Peer validation: The respective peer on the other end of the connection can be verified to be the correct party with whom communication is intended ▪ Downgrade protection: The protocol, if it supports multiple versions, must always use a version both peers agree on and may not be artificially downgraded by an adversary ▪ Replay protection: Data that has previously been recorded by an adversary and that is repeated by that adversary is detected as a duplicate and properly rejected Typically, this is achieved by using TLS as the foundational transport protocol, which, in a correct configuration, can achieve all of these security protocols. Note, however, that even a TLS configuration can be susceptible to attacks on these security goals; most notably if poor choices in the protocol parameterization are used (e.g., weak cipher suites), specific security mechanisms are disabled (e.g., peer validation). Replay protection may be deliberately sacrificed in specific scenarios as well. One example of this would be the use of the 0RTT feature of TLSv1.3. This is permissible if and only if the concerned software has other means of ensuring the replay of messages does not impact the overall security of the system. Specific resource constraints lead to a situation in which deeply embedded devices may not have the resources to fulfill a full TLS handshake; they still need to make sure that the desired security properties are met. *Requirement enhancement 1: L2–L5* For devices that target security level 2 and up, the implementation must either follow an industry-standard security protocol (such as TLS) or a proof of the security properties must be provided that has been vetted by experts in the field. Note that this is typically a task that is exceedingly difficult to achieve because of the required expertise in the field of theoretical cryptography and cryptanalysis. *Requirement enhancement 2: L3–L5* For devices that target security level 3 and up, custom cryptographic constructions are disallowed and industry-standard protocols must be used either way. Furthermore, for these devices, it is required that all secured communication that falls under this clause also achieves Perfect Forward Secrecy (PFS).

14 : Annex J: Mapping to World Wide Web Coalition Web of Things Requirements

The Web of Things (WoT) of the World Wide Web Coalition (W3C) is a set of activities relating the Internet of Things with the objects, models, protocols, standards and best practices of the Web, with an overall goal of reducing fragmentation of the IoT.

CATEGORY	MAPS TO
Secure Device Capabilities – Baseline DEVICE IDENTIFIERS	
Secure Device Capabilities – Baseline SECURED ACCESS	[WOT BP] Recommendations regarding authentication and access control when using HTTPS, CoAPS and MQTTS are at https://w3c.github.io/wot-security-best-practices/#authentication-and-access-control.
Secure Device Capabilities – Baseline DATA IN TRANSIT IS PROTECTED	[WOT BP] Recommendations regarding use of TLS and DTLS when using HTTPS, CoAPS and MQTTS are at https://w3c.github.io/wot-security-best-practices/#secure-transport. [WOT BP] Recommendations regarding end-to-end object protection are at https://w3c.github.io/wot-security-best-practices/#object-security.
Secure Device Capabilities – Baseline DATA AT REST IS PROTECTED	
Secure Device Capabilities – Baseline INDUSTRY ACCEPTED PROTOCOLS ARE USED FOR COMMUNICATIONS	[WOT BP] Recommendations regarding use secure protocols TLS and DTLS are at https://w3c.github.io/wot-security-best-practices/#secure-transport.
Secure Device Capabilities – Baseline DATA VALIDATION	
Secure Device Capabilities – Baseline EVENT LOGGING	
Secure Device Capabilities – Baseline CRYPTOGRAPHY	[WOT BP] See https://w3c.github.io/wot-security-best-practices/ for a discussion of application (HTTPS/CoAPS/MQTTS) and transport (TLS/DTLS) security.

CATEGORY	MAPS TO
Secure Device Capabilities – Baseline PATCHABILITY	[WOT BP] Recommended standards related to patchability are discussed at https://w3c.github.io/wot-security-best-practices/#secure-update-and-post-manufacturing-provisioning.
Secure Device Capabilities – Baseline REPROVISIONING	[WOT BP] Recommended standards related to reprovisioning are discussed at https://w3c.github.io/wot-security-best-practices/#secure-update-and-post-manufacturing-provisioning.
Product Lifecycle Management VULNERABILITY SUBMISSION AND HANDLING PROCESS	
Product Lifecycle Management EOL/EOS UPDATES AND DISCLOSURE	
Product Lifecycle Management DEVICE INTENT DOCUMENTATION	
Secure Capabilities – Phase In Over Time DEVICE INTENT SIGNALING	
Secure Capabilities – Phase In Over Time DEVICE NETWORK ONBOARDING	
Additional IoT Device Security Capabilities and Practices SECURE DEVELOPMENT LIFECYCLE	
Additional IoT Device Security Capabilities and Practices HARDWARE ROOTED SECURITY	
Additional IoT Device Security Capabilities and Practices TIME DISTRIBUTION	
Additional IoT Device Security Capabilities and Practices SYSTEM RESILIENCY	

CATEGORY	MAPS TO
Additional IoT Device Security Capabilities and Practices SECURE TOOLCHAINS	[WOT TP] Discussion of security and tools is at https://w3c.github.io/wot-security-testing-plan/.
Additional IoT Device Security Capabilities and Practices SOFTWARE TRANSPARENCY AND BILL OF MATERIALS	
Additional IoT Device Security Capabilities and Practices LEAST FUNCTIONALITY	
Additional IoT Device Security Capabilities and Practices PHYSICAL ACCESS CONTROL	
Additional IoT Device Security Capabilities and Practices BEST CURRENT PRACTICES	[WOT BP] See generally https://w3c.github.io/wot-security-best-practices/.

15 : Annex K: Mapping to ETSI EN 303 645 (Final Draft, V2.1.0, 2020-04)

CATEGORY	MAPS TO
Secure Device Capabilities – Baseline DEVICE IDENTIFIERS	[EN303645] Provision 5.3-16
Secure Device Capabilities – Baseline SECURED ACCESS	[EN303645] 5.1: No universal default passwords [EN303645] Provision 5.1-2 *through* [EN303645] Provision 5.1-5 [EN303645] Provision 5.4-2 [EN303645] Provision 5.4-3 [EN303645] Provision 5.5-4 [EN303645] Provision 5.5-5 [EN303645] Provision 5.6-1 *through* [EN303645] Provision 5.6-4
Secure Device Capabilities – Baseline DATA IN TRANSIT IS PROTECTED	[EN303645] Provision 5.4-4 [EN303645] Provision 5.5-1 [EN303645] Provision 5.5-6 [EN303645] Provision 5.8-1 [EN303645] Provision 5.8-2
Secure Device Capabilities – Baseline DATA AT REST IS PROTECTED	[EN303645] Provision 5.4-1 *through* [EN303645] Provision 5.4-4 [EN303645] Provision 5.5-7 [EN303645] Provision 5.5-8
Secure Device Capabilities – Baseline INDUSTRY ACCEPTED PROTOCOLS ARE USED FOR COMMUNICATIONS	[EN303645] Provision 5.5-2
Secure Device Capabilities – Baseline DATA VALIDATION	[EN303645] Provision 5.13-1
Secure Device Capabilities – Baseline EVENT LOGGING	[EN303645] Provision 5.7-2
Secure Device Capabilities – Baseline CRYPTOGRAPHY	[EN303645] Provision 5.5-1 [EN303645] Provision 5.3-7 [EN303645] Provision 5.5-3
Secure Device Capabilities – Baseline PATCHABILITY	[EN303645] Provision 5.3-1 *through* [EN303645] Provision 5.3-12 [EN303645] Provision 5.3-15 [EN303645] Provision 5.4-4
Secure Device Capabilities – Baseline REPROVISIONING	[EN303645] Provision 5.11-1 *through* [EN303645] Provision 5.11-4

CATEGORY	MAPS TO
Product Lifecycle Management VULNERABILITY SUBMISSION AND HANDLING PROCESS	[EN303645] Provision 5.2-1 *through* [EN303645] Provision 5.2-3
Product Lifecycle Management EOL/EOS UPDATES AND DISCLOSURE	[EN303645] Provision 5.3-13 [EN303645] Provision 5.3-14
Product Lifecycle Management DEVICE INTENT DOCUMENTATION	
Secure Capabilities – Phase In Over Time DEVICE INTENT SIGNALING	
Secure Capabilities – Phase In Over Time DEVICE NETWORK ONBOARDING	[EN303645] Provision 5.12-1 *through* [EN303645] Provision 5.12-3
Additional IoT Device Security Capabilities and Practices SECURE DEVELOPMENT LIFECYCLE	
Additional IoT Device Security Capabilities and Practices HARDWARE ROOTED SECURITY	[EN303645] Provision 5.7-1
Additional IoT Device Security Capabilities and Practices TIME DISTRIBUTION	
Additional IoT Device Security Capabilities and Practices SYSTEM RESILIENCY	[EN303645] Provision 5.9-1 *through* [EN303645] Provision 5.9-3
Additional IoT Device Security Capabilities and Practices SECURE TOOLCHAINS	
Additional IoT Device Security Capabilities and Practices SOFTWARE TRANSPARENCY AND BILL OF MATERIALS	
Additional IoT Device Security Capabilities and Practices LEAST FUNCTIONALITY	[EN303645] Provision 5.6-5 *through* [EN303645] Provision 5.6-7
Additional IoT Device Security Capabilities and Practices PHYSICAL ACCESS CONTROL	
Additional IoT Device Security Capabilities and Practices BEST CURRENT PRACTICES	[EN303645] Provision 5.6-8 [EN303645] Provision 5.6-9

16 : Annex L: Mapping to ETSI TS 103 645

CATEGORY	MAPS TO
Secure Device Capabilities – Baseline DEVICE IDENTIFIERS	
Secure Device Capabilities – Baseline SECURED ACCESS	[TS103645] 4.1 No universal default passwords [TS103645] Provision 4.1-1 [TS103645] Provision 4.6-1 [TS103645] Provision 4.6-2 [TS103645] Provision 4.6-3
Secure Device Capabilities – Baseline DATA IN TRANSIT IS PROTECTED	[TS103645] 4.4 Securely store credentials and security-sensitive data [TS103645] 4.5 Communicate securely [TS103645] Provision 4.5-1 [TS103645] Provision 4.5-2
Secure Device Capabilities – Baseline DATA AT REST IS PROTECTED	[TS103645] Provision 4.10-1 [TS103645] Provision 4.10-2 [TS103645] Provision 4.10-3
Secure Device Capabilities – Baseline INDUSTRY ACCEPTED PROTOCOLS ARE USED FOR COMMUNICATIONS	[TS103645] 4.5 Communicate securely
Secure Device Capabilities – Baseline DATA VALIDATION	[TS103645] Provision 4.13-1
Secure Device Capabilities – Baseline EVENT LOGGING	[TS103645] Provision 4.7-2
Secure Device Capabilities – Baseline CRYPTOGRAPHY	
Secure Device Capabilities – Baseline PATCHABILITY	[TS103645] 4.3 Keep software updated [TS103645] Provision 4.3-1 *through* Provision 4.3-9
Secure Device Capabilities – Baseline REPROVISIONING	
Product Lifecycle Management VULNERABILITY SUBMISSION AND HANDLING PROCESS	[TS103645] 4.2: Implement a means to manage reports of vulnerabilities [TS103645] Provision 4.2-1 *through* Provision 4.2-3
Product Lifecycle Management EOL/EOS UPDATES AND DISCLOSURE	

CATEGORY	MAPS TO
Product Lifecycle Management DEVICE INTENT DOCUMENTATION	
Secure Capabilities – Phase In Over Time DEVICE INTENT SIGNALING	
Secure Capabilities – Phase In Over Time DEVICE NETWORK ONBOARDING	[TS103645] 4.12 Make installation and maintenance of devices easy [TS103645] Provision 4.12-1
Additional IoT Device Security Capabilities and Practices SECURE DEVELOPMENT LIFECYCLE	
Additional IoT Device Security Capabilities and Practices HARDWARE ROOTED SECURITY	[TS103645] Provision 4.7-1
Additional IoT Device Security Capabilities and Practices TIME DISTRIBUTION	
Additional IoT Device Security Capabilities and Practices SYSTEM RESILIENCY	[TS103645] Provision 4.9-1 *through* Provision 4.9-3
Additional IoT Device Security Capabilities and Practices SECURE TOOLCHAINS	
Additional IoT Device Security Capabilities and Practices SOFTWARE TRANSPARENCY AND BILL OF MATERIALS	
Additional IoT Device Security Capabilities and Practices LEAST FUNCTIONALITY	[TS103645] Provision 4.6-4 [TS103645] Provision 4.6-5
Additional IoT Device Security Capabilities and Practices PHYSICAL ACCESS CONTROL	
Additional IoT Device Security Capabilities and Practices BEST CURRENT PRACTICES	

17 : Annex M: Mapping to EU Agency for Cybersecurity Baseline Security Recommendations for IoT

This section maps this group's recommendations[3] to the C2 Consensus. Note that the EU Agency for Cybersecurity was previously known as ENISA.

CATEGORY	MAPS TO
Secure Device Capabilities – Baseline DEVICE IDENTIFIERS	[ENISA] (Annex A): GP-PS-10
Secure Device Capabilities – Baseline SECURED ACCESS	[ENISA] (Annex A): GP-TM-09 [ENISA] (Annex A): GP-TM-21 *through* [ENISA] (Annex A): GP-TM-27 [ENISA] (Annex A): GP-TM-43
Secure Device Capabilities – Baseline DATA IN TRANSIT IS PROTECTED	[ENISA] (Annex A): GP-PS-10 [ENISA] (Annex A): GP-TM-34 [ENISA] (Annex A): GP-TM-38 *through* [ENISA] (Annex A): GP-TM-43
Secure Device Capabilities – Baseline DATA AT REST IS PROTECTED	
Secure Device Capabilities – Baseline INDUSTRY ACCEPTED PROTOCOLS ARE USED FOR COMMUNICATIONS	[ENISA] (Annex A): GP-OP-04
Secure Device Capabilities – Baseline DATA VALIDATION	[ENISA] 4.3.13 Secure input and output handling [ENISA] (Annex A): GP-TM-54
Secure Device Capabilities – Baseline EVENT LOGGING	[ENISA] (Annex A): GP-TM-55
Secure Device Capabilities – Baseline CRYPTOGRAPHY	[ENISA] (Annex A): GP-TM-35 *through* [ENISA] (Annex A): GP-TM-37
Secure Device Capabilities – Baseline PATCHABILITY	[ENISA] (Annex A): GP-TM-18 *through* [ENISA] (Annex A): GP-TM-20
Secure Device Capabilities – Baseline REPROVISIONING	[ENISA] (Annex A): GP-OP-01 [ENISA] (Annex A): GP-OP-02
Product Lifecycle Management VULNERABILITY SUBMISSION AND HANDLING PROCESS	[ENISA] (Annex A): GP-OP-03 [ENISA] (Annex A): GP-OP-05 *through* [ENISA] (Annex A): GP-OP-08
Product Lifecycle Management DEVICE INTENT DOCUMENTATION	

CATEGORY	MAPS TO
Secure Capabilities – Phase In Over Time DEVICE INTENT SIGNALING	
Secure Capabilities – Phase In Over Time DEVICE NETWORK ONBOARDING	
Additional IoT Device Security Capabilities and Practices SECURE DEVELOPMENT LIFECYCLE	
Additional IoT Device Security Capabilities and Practices HARDWARE ROOTED SECURITY	[ENISA] (Annex A): GP-TM-01 [ENISA] (Annex A): GP-TM-02
Additional IoT Device Security Capabilities and Practices SECURE TOOLCHAINS	
Additional IoT Device Security Capabilities and Practices SOFTWARE TRANSPARENCY AND BILL OF MATERIALS	
Additional IoT Device Security Capabilities and Practices LEAST FUNCTIONALITY	
Additional IoT Device Security Capabilities and Practices PHYSICAL ACCESS CONTROL	
Additional IoT Device Security Capabilities and Practices BEST CURRENT PRACTICES	

18 : Annex N: Mapping to GSMA IoT Security Guidelines for Endpoint Ecosystems

CATEGORY	MAPS TO
Secure Device Capabilities – Baseline DEVICE IDENTIFIERS	
Secure Device Capabilities – Baseline SECURED ACCESS	[GSMA] 6.9 Endpoint Password Management [GSMA] 6.12 Remote Endpoint Administration
Secure Device Capabilities – Baseline DATA IN TRANSIT IS PROTECTED	[GSMA] 6.14 Enforce Memory Protection [GSMA] 6.15 Bootloading Outside of Internal ROM [GSMA] 6.16 Locking Critical Sections of Memory
Secure Device Capabilities – Baseline DATA AT REST IS PROTECTED	
Secure Device Capabilities – Baseline INDUSTRY ACCEPTED PROTOCOLS ARE USED FOR COMMUNICATIONS	[GSMA] 6.19 Endpoint Communications Security
Secure Device Capabilities – Baseline DATA VALIDATION	
Secure Device Capabilities – Baseline EVENT LOGGING	[GSMA] 6.13 Logging and Diagnostics
Secure Device Capabilities – Baseline CRYPTOGRAPHY	
Secure Device Capabilities – Baseline PATCHABILITY	
Secure Device Capabilities – Baseline REPROVISIONING	
Product Lifecycle Management VULNERABILITY SUBMISSION AND HANDLING PROCESS	
Product Lifecycle Management EOL/EOS UPDATES AND DISCLOSURE	
Product Lifecycle Management DEVICE INTENT DOCUMENTATION	

CATEGORY	MAPS TO
Secure Capabilities – Phase In Over Time DEVICE INTENT SIGNALING	
Secure Capabilities – Phase In Over Time DEVICE NETWORK ONBOARDING	[GSMA] 6.8 Uniquely Provision Each Endpoint [GSMA] 6.20 Authenticating an Endpoint Identity
Additional IoT Device Security Capabilities and Practices SECURE DEVELOPMENT LIFECYCLE	
Additional IoT Device Security Capabilities and Practices HARDWARE ROOTED SECURITY	[GSMA] 6.1 Implement an Endpoint Trusted Computing Base [GSMA] 6.2 Utilize a Trust Anchor [GSMA] 6.3 Use a Tamper Resistant Trust Anchor [GSMA] 6.4 Define an API for Using the TCB [GSMA] 6.5 Defining an Organizational Root of Trust [GSMA] 6.6 Personalize Each Endpoint Device Prior to Fulfilment [GSMA] 6.7 Minimum Viable execution Platform (Application Roll-Back)
Additional IoT Device Security Capabilities and Practices TIME DISTRIBUTION	
Additional IoT Device Security Capabilities and Practices SYSTEM RESILIENCY	
Additional IoT Device Security Capabilities and Practices SECURE TOOLCHAINS	
Additional IoT Device Security Capabilities and Practices SOFTWARE TRANSPARENCY AND BILL OF MATERIALS	
Additional IoT Device Security Capabilities and Practices LEAST FUNCTIONALITY	
Additional IoT Device Security Capabilities and Practices PHYSICAL ACCESS CONTROL	
Additional IoT Device Security Capabilities and Practices BEST CURRENT PRACTICES	

19 : Annex O: Mapping to NISTIR 8259/8259A *IoT Device Cybersecurity Capability Core Baseline and Activities*

The [NISTIR 8259A Baseline] and [NIST8259 Activities] documents were released in May of 2020. The Baseline in 8259A represents a core, fundamental set of capabilities that are expected of all connected devices in all sectors. The C2 Consensus is a multi-sector core Baseline that maps onto the NIST Baseline and Activities as shown below.

The mapping shows likely places to explore the commonalities between the C2 Consensus Baseline and the NIST guidance in [NISTIR 8259A Baseline] and [NISTIR 8259A Activities]. Please refer to the NIST documents for details in each section.

CATEGORY	MAPS TO
Secure Device Capabilities – Baseline DEVICE IDENTIFIERS	[NISTIR 8259A Baseline] Table 1 row 1, Device Identification: The IoT device can be uniquely identified logically and physically.
Secure Device Capabilities – Baseline SECURED ACCESS	[NISTIR 8259A Baseline] Table 1 row 2, Device Configuration: The configuration of the IoT device's software can be changed, and such changes can be performed by authorized entities only. Table 1 row 4, Logical Access to Interfaces: The IoT device can restrict logical access to its local and network interfaces, and the protocols and services used by those interfaces, to authorized entities only.
Secure Device Capabilities – Baseline DATA IN TRANSIT IS PROTECTED	[NISTIR 8259A Baseline] Table 1 row 3, Data Protection: The IoT device can protect the data it stores and transmits from unauthorized access and modification.
Secure Device Capabilities – Baseline DATA AT REST IS PROTECTED	[NISTIR 8259A Baseline] Table 1 row 3, Data Protection: The IoT device can protect the data it stores and transmits from unauthorized access and modification.
Secure Device Capabilities – Baseline INDUSTRY ACCEPTED PROTOCOLS ARE USED FOR COMMUNICATIONS	
Secure Device Capabilities – Baseline DATA VALIDATION	[NISTIR 8259 Activities] Section 3.4, Activity 4: Plan for Adequate Support of Customer Needs and Goals (recommendations under "4. What measures are taken to minimize the vulnerabilities in released IoT device software?")
Secure Device Capabilities – Baseline EVENT LOGGING	[NISTIR 8259A Baseline] Table 1 row 6, Cybersecurity Event Logging: The IoT device can report on its cybersecurity state and make that information accessible to authorized entities only.

CATEGORY	MAPS TO
Secure Device Capabilities – Baseline CRYPTOGRAPHY	[NISTIR 8259A Baseline] Table 1 row 3, Data Protection: The IoT device can protect the data it stores and transmits from unauthorized access and modification.
Secure Device Capabilities – Baseline PATCHABILITY	[NISTIR 8259A Baseline] Table 1 row 5, Software and Firmware Update: The IoT device's software can be updated by authorized entities only using a secure and configurable mechanism.
Secure Device Capabilities – Baseline REPROVISIONING	[NISTIR 8259A Baseline] Table 1 row 2, Device Configuration: The configuration of the IoT device's software can be changed, and such changes can be performed by authorized entities only. Table 1 row 4, Logical Access to Interfaces: The IoT device can limit logical access to its local and network interfaces to authorized entities only.
Product Lifecycle Management VULNERABILITY SUBMISSION AND HANDLING PROCESS	[NISTIR 8259 Activities] Section 2 (discussion of NISTIR 8228, with regard to Vulnerability Management)
Product Lifecycle Management EOL/EOS UPDATES AND DISCLOSURE	[NISTIR 8259 Activities] Section 3.2 (2), bullet 6: "Consider expectations about device lifespan…"
Product Lifecycle Management DEVICE INTENT DOCUMENTATION	[NISTIR 8259 Activities] Section 4.2.3 (5): Discussion of information regarding the IoT device's operational characteristics
Secure Capabilities – Phase In Over Time DEVICE INTENT SIGNALING	[NISTIR 8259A Baseline] Table 1 row 1, Device Identification: The IoT device can be uniquely identified logically and physically (last bullet under Rationale)
Secure Capabilities – Phase In Over Time DEVICE NETWORK ONBOARDING	[NISTIR 8259A Baseline] Table 1 row 4, Logical Access to Interfaces: The IoT device can restrict logical access to its local and network interfaces, and the protocols and services used by those interfaces, to authorized entities only (last bullet under Rationale)
Additional IoT Device Security Capabilities and Practices SECURE DEVELOPMENT LIFECYCLE	[NISTIR 8259 Activities] Section 3.4, Activity 4: Plan for Adequate Support of Customer Needs and Goals (discussion of secure development practices)
Additional IoT Device Security Capabilities and Practices HARDWARE ROOTED SECURITY	[NISTIR 8259 Activities] Section 3.4, Activity 4: Plan for Adequate Support of Customer Needs and Goals (discussion of hardware-based security)
Additional IoT Device Security Capabilities and Practices TIME DISTRIBUTION	

CATEGORY	MAPS TO
Additional IoT Device Security Capabilities and Practices SYSTEM RESILIENCY	
Additional IoT Device Security Capabilities and Practices SECURE TOOLCHAINS	
Additional IoT Device Security Capabilities and Practices SOFTWARE TRANSPARENCY AND BILL OF MATERIALS	[NISTIR 8259 Activities] 4.2.3 Device Composition and Capabilities (discussion of information customers need abut the sources of the device's software, hardware and services, especially footnote 6)
Additional IoT Device Security Capabilities and Practices LEAST FUNCTIONALITY	[NISTIR 8259 Activities] Section 3.4, Activity 4: Plan for Adequate Support of Customer Needs and Goals (applies to a limited definition of Least Functionality; see discussion of "unneeded device capabilities")
Additional IoT Device Security Capabilities and Practices PHYSICAL ACCESS CONTROL	[NISTIR 8259 Activities] Section 3.4, Activity 4: Plan for Adequate Support of Customer Needs and Goals (discussion of "tamper-resistant enclosure")
Additional IoT Device Security Capabilities and Practices BEST CURRENT PRACTICES	[NISTIR 8259 Activities] Section 3.4, Activity 4: Plan for Adequate Support of Customer Needs and Goals (discussion beginning with, "Manufacturers should consider which secure development practices are most appropriate...")

20 : Annex P: Mapping to UK DCMS Code of Practice for Consumer IoT Security

https://www.gov.uk/government/publications/code-of-practice-for-consumer-iot-security/code-of-practice-for-consumer-iot-security

CATEGORY	MAPS TO
Secure Device Capabilities – Baseline DEVICE IDENTIFIERS	
Secure Device Capabilities – Baseline SECURED ACCESS	[DCMS] 1. No default passwords
Secure Device Capabilities – Baseline DATA IN TRANSIT IS PROTECTED	[DCMS] 4. Securely store credentials and security-sensitive data [DCMS] 5. Communicate securely
Secure Device Capabilities – Baseline DATA AT REST IS PROTECTED	
Secure Device Capabilities – Baseline INDUSTRY ACCEPTED PROTOCOLS ARE USED FOR COMMUNICATIONS	
Secure Device Capabilities – Baseline DATA VALIDATION	[DCMS] 13. Validate input data
Secure Device Capabilities – Baseline EVENT LOGGING	
Secure Device Capabilities – Baseline CRYPTOGRAPHY	
Secure Device Capabilities – Baseline PATCHABILITY	[DCMS] 3. Keep software updated
Secure Device Capabilities – Baseline REPROVISIONING	[DCMS] 11. Make it easy for consumers to delete personal data.
Product Lifecycle Management VULNERABILITY SUBMISSION AND HANDLING PROCESS	[DCMS] 2. Implement a vulnerability disclosure policy
Product Lifecycle Management EOL/EOS UPDATES AND DISCLOSURE	

CATEGORY	MAPS TO
Product Lifecycle Management DEVICE INTENT DOCUMENTATION	
Secure Capabilities – Phase In Over Time DEVICE INTENT SIGNALING	
Secure Capabilities – Phase In Over Time DEVICE NETWORK ONBOARDING	[DCMS] 12. Make installation and maintenance of devices easy
Additional IoT Device Security Capabilities and Practices SECURE DEVELOPMENT LIFECYCLE	
Additional IoT Device Security Capabilities and Practices HARDWARE ROOTED SECURITY	[DCMS] 7. Ensure software integrity (Software on IoT devices should be verified using secure boot mechanisms....)
Additional IoT Device Security Capabilities and Practices TIME DISTRIBUTION	
Additional IoT Device Security Capabilities and Practices SYSTEM RESILIENCY	[DCMS] 9. Make systems resilient to outages
Additional IoT Device Security Capabilities and Practices SECURE TOOLCHAINS	
Additional IoT Device Security Capabilities and Practices SOFTWARE TRANSPARENCY AND BILL OF MATERIALS	
Additional IoT Device Security Capabilities and Practices LEAST FUNCTIONALITY	[DCMS] 6. Minimise exposed attack surfaces
Additional IoT Device Security Capabilities and Practices PHYSICAL ACCESS CONTROL	
Additional IoT Device Security Capabilities and Practices BEST CURRENT PRACTICES	

21 : Endnotes

1 [NIST SP1800-15A], Executive Summary at 33.

2 Ibid., Table 5-1, page 23.

3 See *EU Agency for Cybersecurity Baseline Security Recommendations for IoT*, https://www.enisa.europa.eu/publications/baseline-security-recommendations-for-iot.

CSDE

Council to Secure the Digital Economy

securingdigitaleconomy.org