

The C2 Consensus on IoT Device Security Baseline Capabilities

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



Council to Secure the
Digital Economy

Acknowledgement

The following organizations contributed technical material, provided ongoing advice and support, and helped craft the recommendations in this document. The editors gratefully acknowledge the contributions of these and other groups:

Alliance for Telecommunications Industry Solutions (ATIS)

Association of Home Appliance Manufacturers (AHAM)

BSA | The Software Alliance

CableLabs

Coalition for Cybersecurity Policy and Law

CompTIA

Consumer Technology Association (CTA)

Council to Secure the Digital Economy (CSDE)

CTIA

Industrial Internet Consortium (IIC)

Information Technology Industry Council (ITI)

Internet of Secure Things (IoXT)

Internet Society

IoTopia

NCTA — The Internet & Television Association

Open Connectivity Foundation (OCF)

Telecommunications Industry Association (TIA)

UL

U.S. Chamber of Commerce (USCC)

USTelecom - The Broadband Association (USTelecom)

LETTER FROM

Gary Shapiro, President and Chief Executive Officer, CTA

Jonathan Spalter, President and Chief Executive Officer, USTelecom

ALONG WITH THE TREMENDOUS BENEFITS that the rapid growth of the Internet of Things (IoT) brings to consumers, businesses, governments and the global digital economy, the IoT's growth also brings increased threats to the digital economy.

This is why the Council to Secure the Digital Economy (CSDE) — composed of USTelecom, the Consumer Technology Association (CTA), and 13 global information and communications technology (ICT) companies — has convened technical experts from 19 leading organizations throughout the ICT sector to develop and advance industry consensus on baseline security capabilities for new devices.

This convening of the conveners — or C2 — has brought together trade associations, standards development organizations, industry alliances and coalitions to develop the C2 Consensus Baseline, the broadest and most technically deep industry consensus on IoT security worldwide. This effort is based on the principle that the best way to achieve IoT security is for technical experts to develop and advance security specifications that will spread throughout the global market.

This document provides clear expert guidance to industry and government on securing new IoT devices in order to raise the market's expectations for security and to advance global policy harmonization. It is our expectation that this global approach will prove more effective than disparate local initiatives that would fragment security requirements and cause inefficiencies in the market that result in weaker security.

We thank all C2 participants, which collectively represent thousands of companies and many different segments of the global digital economy, for their engagement and their valuable contributions.

We look forward to promoting the C2 Consensus Baseline in key venues around the world to move the global market for IoT toward security.

Sincerely,



Gary Shapiro
President and CEO, Consumer Technology Association



Jonathan Spalter
President and CEO, USTelecom



Contents

| | | |
|-----------|--|----|
| 01 | Foreword | 4 |
| 02 | Definitions and Acronyms | 5 |
| 03 | Background | 6 |
| 04 | Methodology | 9 |
| 05 | Consensus Baseline IoT Device Security Capabilities | 10 |
| | 5.1. <i>Secure Device Capabilities – Baseline</i> | 10 |
| | 5.1.1. Device Identifiers | 10 |
| | 5.1.2. Secured Access | 12 |
| | 5.1.3. Data In Transit Is Protected | 13 |
| | 5.1.4. Data At Rest Is Protected | 14 |
| | 5.1.5. Industry Accepted Protocols are Used for Communications | 15 |
| | 5.1.6. Data Validation | 15 |
| | 5.1.7. Event Logging | 16 |
| | 5.1.8. Cryptography | 16 |
| | 5.1.9. Patchability | 17 |
| | 5.1.10. Reprovisioning | 18 |
| | 5.2. <i>Product Lifecycle Management Capabilities - Baseline</i> | 18 |
| | 5.2.1. Vulnerability Submission and Handling Process | 18 |
| | 5.2.2. EoL/EoS Updates and Disclosure | 19 |
| | 5.2.3. Device Intent Documentation | 19 |
| 06 | Annex A: Regarding Future Secure Capabilities – Phase in Over Time | 21 |
| | A.1 <i>Device Intent Signaling</i> | 21 |
| | A.2 <i>Device Network Onboarding</i> | 21 |
| 07 | Annex B: Additional IoT Device Security Capabilities and Practices | 23 |
| | B.1. <i>Secure Development Lifecycle</i> | 23 |
| | B.2. <i>Hardware Rooted Security</i> | 23 |
| | B.3. <i>Time Distribution</i> | 24 |
| | B.4. <i>System Resiliency</i> | 24 |
| | B.5. <i>Secure Toolchains</i> | 25 |
| | B.6. <i>Software Transparency and Bill of Materials</i> | 25 |
| | B.7. <i>Least Functionality</i> | 26 |
| | B.8. <i>Physical Access Control</i> | 26 |
| | B.9. <i>Best Current Practices</i> | 26 |



| | | |
|-----------|---|----|
| 08 | Annex C: Discussion of Implementation and Complexity | 27 |
| 09 | Annex D: Informative References | 30 |
| 10 | Annex E: Mapping to CSDE International Anti-Botnet Guide..... | 31 |
| 11 | Annex F: Mapping to CTIA IoT Device Cybersecurity Certification | 34 |
| 12 | Annex G: Mapping to IoTopia Specifications..... | 37 |
| 13 | Annex H: Mapping to IoXT Pledge | 41 |
| 14 | Annex I: Mapping to Open Connectivity Foundation Specifications | 44 |
| 15 | Annex J: Mapping to World Wide Web Coalition Web of Things Requirements..... | 47 |
| 16 | Annex K: Mapping to EU Agency for Cybersecurity Baseline Security Recommendations for IoT..... | 49 |
| 17 | Annex L: Mapping to ETSI 103 645 | 54 |
| 18 | Annex M: Mapping to GSMA IoT Security Guidelines for Endpoint Ecosystems..... | 57 |
| 19 | Annex N: Mapping to Draft NISTIR 8259..... | 60 |
| 20 | Annex O: Mapping to UK DCMS Code of Practice for Consumer IoT Security | 63 |
| 21 | Annex P: Mapping to UL MCV 1376 – Security Capabilities Verified | 66 |
| 22 | Sponsoring Organizations | 75 |
| 23 | Endnotes | 76 |



01 | Foreword

THE CONVENE THE CONVENERS (C2) PROJECT coalesces the expertise of hundreds of technical experts via their various conveners: trade associations, standards development organizations, industry alliances and coalitions. The C2 Consensus was developed by many organizations working together on an equal basis to find common ground on IoT device security for new designs. The convening—bringing together—of these groups allowed for sharing and comparing the expert recommendations each had developed within their own constituency. The work was coordinated under the auspices of the Council to Secure the Digital Economy and the Consumer Technology Association.

This is a technical document. Beyond the general technical security principle that the best path to IoT security is for technical experts to develop and advance technical security specifications, any questions of law, regulation, and policy pertaining to data security and privacy are out of scope for this document. (Where the term “policy” is used, it is intended to reference technical and operational policies rather than, for instance, regulatory policies.)

Although the contributors to the C2 Consensus recognize that the security of the installed base of legacy devices is important, this document applies to new device designs.

It is important to note that “consensus” is not a synonym for “unanimity”. Where there was not perfect agreement among C2 participants, the key pros and cons of certain recommendations are captured here.

It is also important to recognize that this Consensus document does not replace or supersede the security work done by these organizations. Each technical document that was used to draft this Consensus document has its place in the IoT world and should be considered on its own merits and in its own context.

02 | Definitions and Acronyms

| | |
|----------------------|---|
| Configuration | The device data related to device identity, credentials and associated data that support that identity |
| Credential | Evidence that supports a claim of identity.* |
| Cryptographic | |
| Certificate | A cryptographically signed structure that binds public keys to an identifier for the entity (i.e., a distinguished name). |
| Device | An entity with one or more endpoints. |
| Endpoint | An entity comprised of one or more components, addressable on a network. |
| Entity | An item with a recognizably distinct existence.† |
| EoL | End of Life (of an IoT device) |
| EoS | End of Service (of an IoT device) |
| Identity | An inherent property of an entity that distinguishes it from all other entities; an identity must exist in a namespace to allow it to be referred to without ambiguity.‡ |
| IoT | Internet of Things. An IoT system involves a physical device that connects to a switched or wireless network, for the purposes of access and control. IoT systems may be connected to open networks, such as the Internet, or closed private networks. An IoT device may have supplementary functions provided through remote execution such as an application running on a phone, tablet, local or 'cloud' based computing system. |
| Managed | (Of environments), supported by trained staff (beyond manufacturer technical support), such as in a large enterprise or in a government office. Compare with unmanaged environment. |
| PKI | Public Key Infrastructure |
| Policy | Policy refers to applicable laws, regulations, and corporate policy. |
| Post-market | After release of the individual device to the field (i.e., after it leaves the factory and goes into the distribution channel). Compare to pre-market. |
| Pre-market | Prior to release of the individual device to the market (e.g., before it leaves the factory and goes into the distribution channel). Compare to post-market. |
| Root of Trust | (Also RoT) A component that performs one or more security-specific functions, such as measurement, storage, reporting, verification, and/or update. |
| Unmanaged | (Of environments), not supported by the owning organization's staff, such as a consumer home or some small businesses. |

* CNSSI 4009, available at <https://rmf.org/wp-content/uploads/2017/10/CNSSI-4009.pdf>

† ISO/IEC 24760-1:2011, available at <https://www.iso.org/standard/57914.html>

‡ ISO/IEC/IEEE 31320-2:2012, available at <https://www.iso.org/standard/60614.html>



03 | Background

IoT Growth and Security

IT HAS BEEN WELL-DOCUMENTED that the Internet of Things (IoT) is growing quickly; the rapid deployment of connected devices is estimated to reach 20 billion units by 2020.¹ This incredible growth is fueled by falling integration costs, explosion of use cases and ubiquitous connectivity.

IoT adoption brings new technologies to many verticals, including manufacturing, medical, automotive, and consumer. In many cases these new technologies may also offer greater robustness, safety, reliability, and resilience by allowing the device to have real-time updates to address security flaws, whereas prior to this technology, such updates were not possible. Still, these technologies often introduce new concerns regarding the safety, reliability, security, resilience, and privacy of the device, leading to potential reduction in the overall trustworthiness of the system. Security is the common denominator across these inter-related trustworthiness disciplines and the security assures they operate as intended. Securing the IoT is a multifaceted challenge that demands a layered approach: security must be addressed not only in IoT devices but also in the network infrastructure, cloud architectures, edge providers, and other elements of the IoT that interact with those devices. Nevertheless, the trustworthiness of the IoT begins with secure devices.

Therefore, while this document acknowledges the relevance and importance of the related trustworthiness disciplines, it focuses exclusively on the security aspects of IoT devices themselves.

The security challenges of the IoT are also well-publicized. Botnets have become particularly and increasingly damaging and costly; they propagate malware,² conduct denial of service attacks,³ and spread disinformation on social media.⁴ A single botnet can now include more than 30 million “zombie” endpoints and allow malicious actors to profit six figures per month.⁵

Aside from being compromised at scale to form botnets, poorly secured IoT devices can be compromised to send spam, secretly collect user data, and hijacked for malicious remote control. Due to the critical nature of many of these devices, as well as the potential to use them as launch points for more damaging attacks, the Center for Strategic & International Studies (CSIS) released a report emphasizing the need for the federal government to consider cybersecurity as a key pillar in its procurement and use of endpoint devices, inclusive of IoT.⁶

Many Efforts, Many Standards

Industry is moving to lock down the IoT in a variety of ways. These methods include software and hardware wrappers to thwart device attacks, and modified traditional security appliances such as firewalls and IDPS specifically designed to focus on the IoT.⁷

While individual industry segments work on security, broad efforts are underway to address this challenge in a harmonized fashion. These efforts are occurring in all parts of the globe. Regulators and other government agencies in many parts of the world have established or are establishing recommendations and requirements, assessment structures and labeling programs.

- 
- ▶ **European Union:** The Cybersecurity Act⁸ will, among other things, allow the EU Agency for Cybersecurity (formerly the EU Agency for Network and Information Security, or ENISA) to set certification schemes for ICT products, services, and processes, to include the IoT.
 - ▶ **Japan:** The Ministry of Economy, Trade and Industry (METI) is developing a Cyber/Physical Security Framework⁹ pertaining to the security of IoT and other connected systems.
 - ▶ **Singapore:** The Infocomm Media Development Authority is developing an IoT Cyber Security Guide.
 - ▶ **United Kingdom:** The Department for Digital, Culture, Media and Sport is active on these issues, for instance issuing a Code of Practice for Consumer IoT Security¹⁰ and recommending regulations to require consumer IoT devices incorporate at least minimum security controls.¹¹
 - ▶ **United States:** The National Institute of Standards and Technology (NIST) has established the Cybersecurity for IoT Program.¹²

Civil society groups are working in similar directions¹³. And of course, industry organizations—trade associations, standards development organizations, industry alliances and coalitions—have crafted a variety of voluntary consensus standards and “best practice” documents for securing IoT devices.¹⁴

Some of these industry documents are best used in a specific context. They may be aimed at vertical markets such as the smart home or medical device markets; or have other contextual boundaries. Other documents are intended for horizontal market application. They are independent of specific application.

The Need for a Common Baseline

Each industry group makes an important contribution in their space and in general, by convening technical experts to build well-thought-out and effective recommendations. But the multiplicity of expert recommendations does create questions about where to start, how to consider such a wealth of overlapping recommendations, and which ones to follow.

There is a need for a common baseline of security capabilities for all IoT devices. Recommendations and requirements for such capabilities that are in place and under development are fragmented. Bringing consensus and harmonization to the current fragmentation will increase the market’s ability to promote IoT security by creating efficiencies of scale in development, manufacturing, support, training, assessment and identification of IoT products with increased security controls.

C2: Convening the Conveners

The C2 project is convening the leveraged expertise of hundreds of technical experts via their conveners: trade associations, standards development organizations, industry alliances and coalitions. The participants and contributors to the C2 process have worked to compare their own technical specifications to those of other such groups.

Each group represents anywhere from dozens to thousands of companies. A number of the groups are international in scope. The technical expertise in the Consensus is informed, therefore, by a global legion of

industry security professionals. The Consensus cannot capture the perspectives and capabilities of all parts of the IoT ecosystem, but it recognizes a few key baselines that can be commonly pursued and flexibly implemented by manufacturers and others that are looking for guidance.

The Consensus articulates the accepted commonalities in IoT device security and also identifies the areas where consensus has not yet developed. In the latter cases, this document notes why consensus is lacking, in what context it may be found, and in some cases offers suggestions about how to achieve complete consensus on such items.

Application of the Consensus

The Consensus Baseline IoT Device Security Capabilities (the “baseline”) is a common set of device security capabilities that can be applied to all new IoT devices that connect to the internet. The baseline is a set of best-practice capabilities that are broadly applicable—vertically and horizontally—across markets. It applies to the diverse range of new IoT devices, accommodating the broad spectrum of device complexity, regardless of the deployment environment. The baseline is intended to be flexible and not prescriptive. Depending on a variety of factors—from device complexity, deployment environment (managed or unmanaged), risk profile, use case and context—the security capabilities outlined in the baseline can be achieved in a variety of ways, with the key being that the ultimate baseline capability is achieved in a way that is applicable to the specific device. For example, a connected dog collar has a different risk profile than a device that is part of an industrial IoT system; the dog collar is less complex and is likely not part of a managed deployment. Both devices should be secure and meet the common set of security capabilities set forth in the baseline, but *how* to meet each capability will vary, just as the risk profile of IoT devices varies.

Likewise, the baseline is a starting point for IoT device security that will need to evolve over time based on both changes in technology and changes to the threat landscape. This document is intended to inform further work on capabilities for IoT device cybersecurity that is more targeted to specific verticals, device types, use cases, etc.

The baseline is also intended to contribute towards government IoT security efforts. For example, the United States Department of Commerce NIST Cybersecurity for IoT Program is in a public process of developing IoT device baseline capabilities that are informed, in part, by NISTIR 8228, *Considerations for Managing IoT Cybersecurity and Privacy Risk*,¹⁵ and Draft NISTIR 8259, *Core Cybersecurity Feature Baseline for Securable IoT Devices: A Starting Point for IoT Device Manufacturers*.¹⁶ C2 participants expect that this Consensus will contribute to that process. More broadly, the global nature of the participants in C2 means that the Consensus should be interesting and important in many jurisdictions, industries, and vertical markets.

The purpose and benefits of this Consensus are two-fold and mutually reinforcing. First and most important, advancing toward an industry consensus security baseline will promote IoT security throughout the global market. The baseline will help to lift *all* new IoT devices’ cybersecurity; the fact that most of the IoT market is comprised of low- or medium-complexity devices¹⁷ makes it all the more important for the baseline to be applicable to low- and medium-complexity devices, and not tailored for high-complexity devices. Second, consensus in industry will streamline and strengthen government-industry collaboration on these issues, allowing for more effective IoT security policies worldwide — and thus bring further improvements in IoT security.



04 | Methodology

CSDE BEGAN THIS PROCESS by surveying the IoT device security capabilities recommendations and voluntary consensus standards available. Those groups that had already contributed important work were identified in a linear list. The list included groups from industry, government and civil society. This initial enumeration amounted to dozens of organizations, too many for a realistic process of consensus-building.

In order to manage the effort, the C2 project team identified a subset of these organizations as potential sources of technical recommendations for a consensus process. Most are from industry. Government and civil society groups have technical expertise as well, of course, but the industry groups leverage the in-house subject matter experts as well as the pragmatic capabilities of the engineers who are building products. Further, government recommendations often follow a public consultation model, bringing in the same industry experts to comment on proposals.

Once the C2 organizations were assembled, each was asked to submit proposed IoT device security capabilities in a standardized format.

The group met March 21st 2019 to compare the data and identify common capabilities. thirteen consensus capabilities were identified. These are identified and explained in Section 5, *“Consensus Baseline IoT Device Security Capabilities”*. As guidance for the future, selected capabilities that are important but still emerging are shown in *“Annex A: Regarding Future Secure Capabilities — Phase In Over Time”*. Those requirements that did not, in the opinion of the group, achieve consensus status are listed in *“Annex B: Additional IoT Device Security Capabilities”*.

“Consensus” in this process did not always represent unanimity but always required a significant majority. Where there was not unanimity, the counterpoints are included in Section 5, *Consensus Baseline IoT Device Security Capabilities*, along with the majority points.

Finally, this is a technical document. Discussions of law, regulation and future law and regulation on data security and privacy are out of scope for this document. Where the term “policy” is used, it is used to clarify application of technical topics. Generally, “policy” here will refer to applicable law, applicable regulation, and corporate decisions (“corporate policy”) about the handling of material. Cybersecurity, as an engineering practice, protects the confidentiality, integrity and availability of that which policy determines is to be protected.



05 | Consensus Baseline IoT Device Security Capabilities

THIS SECTION IDENTIFIES specific “core” baseline security capabilities applicable to all IoT devices.

The first subsection of the baseline, Section 5.1 identifies *device capabilities*. Device capabilities are tangible, testable and verifiable mechanisms built into IoT devices. Broader organizational capabilities for a secure lifecycle are identified in Section 5.2. A third category, secure development, and more generally, organizational cybersecurity risk management practices are beyond the scope of this document.

Some items were identified as important but there was not consensus that the enabling technology was well-enough adopted or developed. Because these items were deemed significant over time, they are included in Annex A as “phase in over time” topics, and should be reviewed by developers for possible future inclusion.

5.1 SECURE DEVICE CAPABILITIES — BASELINE

This section includes device capabilities that are properties of the hardware and software, as opposed to business or development processes or capabilities.

5.1.1 Device Identifiers

Definition: A unique value associated with the endpoint (or values associated with the functional entities within the endpoint) that exists in a namespace to allow it to be referenced without ambiguity. This value is distinct and distinguishes a device from all other devices.

Scope: Identity in the context of device authentication, authorization and management

Discussion: The goal of this capability is to utilize identify information to identify and differentiate a device on the network.

Identity is represented by a single or multiple identifiers. Identifiers play a critical role for IoT security. They are used to address functions and attributes of an IoT device as unique instances which can then be accessed, operated and managed. Identifiers are not just hardware based but can be used for applications and other IoT entities.

Identities play a role across the entire device lifecycle. Identifiers are used to onboard devices to a network(s), register, authenticate, authorize, assign access lists and policy, control and manage the device in the performance of services and applications. Identifiers are used to enumerate the network and identify devices that are or are not intended to be on the network and help trace issues in the event of a breach. Identifiers must be unique, stored and protected. Note that a single device may have multiple entities within the device. These entities could be sub-systems, applications and or services.



It should be noted that the security benefit of these identifiers can be bolstered by additional cryptographic protections for confidentiality, integrity and availability.

For low-end devices, a simpler identifier may suffice to achieve this capability. For example, some resource constrained devices may not be able to mutually authenticate with another device, sign/verify a digital signature, etc. In these cases, the device should be designed to implement as much security as is feasible. In some cases, this may simply consist of not storing data that is not protected.

Multiple Identifiers, examples

An IoT device may have one identifier or a number of different identifiers that may be established at manufacturer or added prior to deployment. Identifiers can be used as part of the device onboarding process, or as part of ongoing device/application management. Each identifier must be unique in a namespace to allow it to be referenced without ambiguity. These identifiers link to various device identities needed for proper authentication and authorization of various functions for device operation and management. Note that in some cases device identity can be added, updated or changed post manufacture or deployment by authorized access.

Examples include:

- ▶ **Device specific, embedded identifiers** associated with the physical hardware of a device, such as Layer 2 MAC addresses used to identify the device to an access network, or the International Mobile Equipment Identity (IMEI) or Mobile Equipment Identifier (MEID) of a cellular device.
- ▶ **Subscription based identifiers** that may be used to enable device access to WAN based network services. These include mobile International Mobile Subscriber Identity or IMSIs used for cellular network access.
- ▶ **IoT application identifiers** that allow an IoT application to identify and access devices for use. Each IoT application using a specific IoT device may have its own unique application identifier.
- ▶ **IoT device management system identifiers**, separate unique identifiers for management access to the devices under the management system's control or scope.
- ▶ **Asset tracking identifiers** such as Electronic Product Codes (EPC) and Tag Identifiers (TID); these are used to obtain track and trace information
- ▶ **Trusted certificates**, which may have a "Unique Name" that is different from all the above but should correlate to a known identity.

Identity/Identifier uses:

Identity is the basis for trustworthiness. Each device should be able to generate, and/or store at least one identifier tied to Identity. The device identity is the building block upon which a broad range of security controls and device manageability depend for proper functionality.

Storage and usage of each of the device identifiers should be protected as appropriate for that identifier. For example, identifiers specific to the physical hardware should be saved in immutable storage components in the device. Provisionable identifiers should also be protected from unauthorized access, changes, and hacks.

Examples of how Identifiers are used in Root of Trust

A Root of Trust (RoT) is a component that performs one or more security-specific functions, such as measurement, storage, reporting, verification, and/or update. These devices/functions are ideally implemented in hardware, are tamper resistant, and create a walled off crypto compute environment that is only accessible via APIs from the device's general compute. For example, a unique secret key might be provisioned into the hardware “root of trust” function which is operated on by the isolated crypto functions. The IoT device never directly sees the underlying secret and never does any of the crypto processing itself.

The highest level of trust that a device can attain depends on the strength of the root of trust. The root of trust, or multiple roots of trust, in a device consist of hardware, software,¹⁸ and other aspects that establish the confidence in the identity of the device or its services. By definition, all devices have one or more roots of trust, and the strength of the root of trust determines the level of confidence in the authenticity of the identifier(s). Much of security is dependent upon the authenticity of device or application identities. The requirement for the strength of the root of trust depends on the threat model and criticality of the device; critical devices may require a hardware root of trust while non-critical devices may suffice with software (or even less capable) root of trust.


Therefore, the hardware RoT is not a panacea, nor is it a viable solution in all devices. On the other hand, without secure storage with crypto capabilities one is effectively hanging the key on a hook next to the lock, so key storage requirements should be considered carefully. See also Section 5.1.4, *Data At Rest is Protected* regarding storage protection requirements.

It should be noted that characteristics that are unique to the device, including Device Identity, can have unintended or undesirable consequences when readable from a distance or over the internet. An example of such an undesirable capability is cyberstalking. Such a features should be configurable or replaceable by a person with local control of the device, particularly if the device can be potentially resold (see also Section 5.1.10 *Reprovisioning*).

5.1.2 Secured Access

Definition: Protection of device operational and management capabilities by requiring user¹⁹ 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.

Scope: This capability includes authenticating authorized users for remote or local access to the operational and management capabilities (including software, firmware and configuration). Authentication may take different forms based on the risk profile of the device, and will depend on the application. It may include requiring a secure certificate from a trusted source, user credentials, biometrics, and/or multi-factor authentication. Authentication must follow good cyber hygiene practices, for example, prevention of default password abuse, device-unique passwords, rate-limiting on password attempts, first-time-change requirements on default passwords, and protection of stored credentials. Credentials should not be shared between users (i.e. there should a unique set of credentials for each user identity that is authenticated).



This capability also includes securing physical interfaces (e.g., debug ports or JTAG) as needed to ensure protection of the software, firmware and configuration. This capability does not include preventing or detecting physical access to the device.

Discussion: This item is intended to help protect the device software, firmware and configuration from unauthorized access either remotely or when a malicious actor has physical access. Note that some devices may not implement an administrative access feature, in which case access may be considered “secured” by this design choice.

A further note with regard to default credentials (e.g. default passwords or a shared certificate); when a non-unique default credential is provided, it should be required to be changed upon first use, or may not be used to provide modification of sensitive parameters.

5.1.3 Data In Transit Is Protected

Definition: Protection of the confidentiality and integrity of selected categories of transmitted data via sound cryptographic means, e.g., HMACs, TLS/DTLS, IPsec, or SSH.

Scope: This capability involves certain data exchanged between the device and other devices, gateways/hubs, and the general internet. An important element of scope is the selection of which data is to be protected; the selection is use-case-specific and should be based on a risk assessment for the device and usage. Note that the user may have some control (settings) over what and whether data is protected in transit.

Discussion: Some devices gather data of little importance or with little temporal value; not all data needs to be protected. Protection may also imply different security properties. For example, confidentiality may be paramount for some data while integrity may be more important for something else. The need and type of protection may be determined based on the data being collected, the context of collection, and other risk factors.

Regardless of data handling policies, certain classes of data should always be protected. For example, data related to the security of the device or system, such as identity and credentials that support that identity (i.e. the configuration) should not be communicated in the clear. Additionally, updates to the software and firmware should also be protected.

Some devices only have internet access via a hub. In those cases, it is important to consider the security of the hub itself, because if the hub does not have baseline security capabilities the device is effectively open to compromise via the hub. It is also important to note that there are low-level or low-capability devices that have limited resources; however, some level of data protection may need to be implemented.

It should be noted that certain types of information—sometimes referred to as “personal”, “sensitive”, or “personally identifiable” information—are subject to rules regarding protection under applicable law and regulation and therefore must be evaluated for protection under those legal frameworks. However, defining “personal”, “sensitive”, or “personally identifiable” information and the required protections are beyond the scope of this document.

Unprotected passwords and unprotected cryptographic keys must always be protected when sent over a public or shared medium.

5.1.4 Data At Rest Is Protected

Definition: Protection of the confidentiality and integrity of selected categories of stored data via sound cryptographic means.

Scope: Data that is stored on the device that, if compromised, would enable attacks at scale such as botnet attacks.

Discussion: This topic requires a certain amount of balance. While the most conservative approach would be to declare all data as worthy of protection without exception, the extreme challenge of such a broad requirement indicates that a use-case evaluation is more appropriate at this time.

While other security concerns should be addressed, it is appropriate to deal with the largest problems—attacks at scale—with the greatest priority. The possibility that a device model can be widely deployed and widely compromised for massive DDoS attacks and social media campaigns is a critical risk.

The possibility that a device model could be compromised is independent of industry and application. Therefore, it is appropriate to discuss this as a baseline capability but also to limit the application of this capability to those devices that have the capacity to be compromised at scale and used in a botnet.


This capability applies to devices with the following key characteristics:

1. The device can be communicated with via the general internet, including behind NAT, using well-known internet protocols
2. The device has an expected useful life (when connected) more than a few days or weeks (this criterion is intended to exclude devices with a very short post-market lifetime, such as package delivery tracking smart labels or tags).

If the device has the above key characteristics, cryptographic measures must be taken to ensure protection of data that, if compromised, would enable attacks at scale such as botnet attacks. Such protective measures should include ensuring the integrity of stored code. Code insertion by an attacker is a common technique of botnet infections and other attacks at scale.

System credentials or keys, user credentials and user data stored on the system should be protected for confidentiality when compromise of this data facilitates attacks at scale. Requirements for user credentials noted in Section 5.1.2 *Secured Access* provide guidance to ensure that user and system credentials are device unique and as such, mitigate their use in such attacks

When protecting the integrity or confidentiality of data, Section 5.1.1 *Device Identifiers* addresses the topic of the RoT which should be considered when implementing Data-at-Rest protection.



Elements of Data-at-Rest may or may not be subject to law or regulation; this must be determined by the type of data stored.

5.1.5 Industry Accepted Protocols are Used for Communications

Definition: Use of secure, widely used protocols, excluding deprecated and replaced versions and protocols, for communications to and from the device.

Scope: Protocols used in exchanging data between the device and other devices; cryptographic standards may sometimes be thought of as “protocols” but are considered separately (see Section 5.1.8, *Cryptography*).

Discussion: “Secure” here means that the protocol does not have a known vulnerability with a ready exploit. Another way to view this item is that it is about the use of security-aware and security-capable protocols for communications to and from the device. But it is important to recognize that even some traditionally accepted protocols may be deprecated now.

Therefore care must be taken in evaluating protocols used. As an example of an insecure implementation, one might “use” TLS 1.3²⁰ as the transport layer security but allow negotiation to settle on SSL 2.0, which has known vulnerabilities and is deprecated by the IETF²¹. Any such fallback must not result in the use of deprecated protocols.

One may also seek “accredited” protocols or “voluntary consensus standards”. In some countries, laws and regulations may limit some options, however. Open, published and peer-reviewed protocols may not be accredited voluntary consensus standards, but at least have had their details reviewed by experts. Where feasible, of course, international or regional voluntary consensus standards are generally best.

5.1.6 Data Validation

Definition: Parsing and limiting input data to prevent it from being used directly as code, commands, or other execution flow inputs; and encoding output data in a form appropriate to and limited to its intended usage.

Scope: This capability applies whenever a device accepts user input, for example for human-readable text fields in a management console.

Discussion: This capability is intended to prevent the large category of exploits that may be available when input data includes special characters or otherwise is conditioned to abuse the data handler. One common type of such exploit is the cross-site scripting (XSS) exploit.²²

For example, when restoring configuration settings to a device by uploading a saved configuration state, the file “../permissions.bin” might be uploaded to overwrite access parameters; stripping “special” characters including the slash is a form of data validation.

Not all devices have data-handling features that would make data validation appropriate, but if they do include such a feature (such as a web interface for configuration), this is a large area for potential abuse.

Note that Section 5.1.2 *Secured Access* has some overlap in intent but has a separate scope.

5.1.7 Event Logging

Definition: A limited persistent record in the device of relevant events, secured and available to authorized users.

Scope: This capability has to do with recording attempts to access the device configuration and other relevant security events. A device needn't keep an infinite number of records and may make use of a simple ring buffer depending on storage limitations. "Relevant events" are device-specific but may include detection of incorrect boot time, failed hash check, or excessive failed login attempts.

Where the device uses a hub or gateway to connect to the internet, the hub or gateway may provide this capability on behalf of the device. This capability is conditioned on the assumption that there is a reasonable likelihood of log inspection for the device type.

Discussion: Logging is a basic need both for forensic analysis, and for real time understanding of system failures. When something goes wrong, it is important to understand what chain of events led to a failure, and what devices are impacted. Logging to an external system is desirable, but not required. Use of standards such as syslog limits storage requirements. However, any mechanism that can provide some indication of anomalous behavior to the administrator — either in real time or retrospectively — is desirable.


5.1.8 Cryptography

Definition: Where cryptography is used, use open, published, proven, and peer-reviewed cryptographic methods with appropriate parameter, algorithm and option selections.

Scope: The technical means used to ensure confidentiality, integrity, and authenticity of data; the technical means used to verify authorization and ensure non-repudiation.

Discussion: Do not implement "home-grown" cryptography. Good cryptography is difficult. It is considerably more difficult when using proprietary solutions. Cryptographic methods should be chosen to match the assessed risk but should use open, proven, peer-reviewed methods and algorithms with—ideally—updateability²³ or the ability to use new cryptographic algorithms.

The purpose of cryptography is to ensure confidentiality, integrity and availability. Example uses may include protecting data in transit (outside the device and in certain cases within the device), protecting data at rest, authentication, authorization, etc. Determining the data to be protected requires some judgement; see related



sections. However, examples of such data may include sensitive data (credentials, etc.) and user defined data (PII, access credentials, etc.)

Note that in some areas the cryptographic methods may be limited to a certain approved set. Within that approval space, the developer should use the best available.

5.1.9 Patchability

Definition: The ability to verifiably update a device’s software and firmware, post-market, with patches that are authenticated to ensure that they have been deployed by an authorized entity as well as to verify the integrity of the patch.

Scope: The patchability capability will vary with device complexity, manageability, and use case. For example, it may not be necessary for all devices to support download of software patches from a remote location; however, such a capability may be the most feasible approach to patch management for all device categories.

Note that some IoT devices are designed to be useful for very short periods of time, after which their purpose is complete and they are removed from service. Examples of such throw-away devices might include disposable smart shipping labels and disposable smart medical bandages.

For such devices, exploits should be patchable pre-market and applicable company policy should determine effective mitigations post-market. To further limit the risks posed by un-patchable “throw-away” devices, the device provider should have a mechanism to identify vulnerable devices, disable vulnerable devices, and communicate the need for replacement of vulnerable devices to end-users.

Note that acknowledgement of such “throw-away” devices does not provide an option to omit patchability by simply declaring a device to have a short lifetime. The patchability capability is intended to be for a reasonably useful period post-market.

Discussion: This capability can be quite difficult from a technical and feasibility point of view. However, it is clear that patchability is necessary in today’s world, unless the device will be taken offline or decommissioned when an update is not possible. Over-the-wire or over-the-air and automated patching for connected devices is preferred to more manual means.

Devices should have the ability to validate patches and ensure that they are unmodified and have not been tampered with. The patch should not reset the settings of the IoT device. Where feasible, using a cryptographic data origin authentication mechanism (e.g., a digital signature or (H)MAC) to protect the patch and validate that it has not been modified is appropriate. Application (or code) signing, where applicable, should also be considered.

5.1.10 Reprovisioning

Definition: The ability for authorized users 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.

Scope: This capability applies to the device configuration, including the initial “as-shipped” configuration, any additional pre-set configurations available to users, and the “as-used” configuration after the device is deployed. See the definition of “configuration” in Section 2, *Definitions and Acronyms*.

Discussion: In the Definition of this capability, the phrase “securely remove” does not have a widely agreed upon definition, and may vary; it may be an action defined by organization policy commensurate with risk that may leave the device in a default/factory-fresh state or other defined state.

Note that, depending on device hardware details, simply wiping memory may or may not be sufficient. Or it may be sufficient to erase memory allocation tables in some devices, but not in others.

Although use of a ‘reset command’ may allow for the easy reset of a system, the implementation of such a command may allow for remote denial of service attacks, or similar exploitations. Therefore, consideration of the risk environment of the system must be made prior to deploying any such solution. A device capability to restore to factory settings is appropriate and should have multiple security protections for managed IoT devices deployed at scale (e.g., smart city deployments) and support the corresponding protection mechanisms.

5.2 PRODUCT LIFECYCLE MANAGEMENT CAPABILITIES - BASELINE

This section considers the important capabilities that are in scope for the organization, rather than the device. Device capabilities are typically observable on a given device. These product lifecycle management capabilities are activities of the manufacturing organization (or otherwise responsible development organization) that are important in the context of overall security of the device.

5.2.1 Vulnerability Submission and Handling Process

Definition: A defined and managed process for accepting vulnerability notifications and acting on them.

Scope: This is a business and engineering process capability for handling information related to software vulnerabilities, interacting with internal staff and external parties who are part of that information flow, and actually addressing the vulnerabilities themselves.



Discussion: The capability to handle vulnerabilities does not imply transparency. Vulnerability transparency is a policy or management action regarding notifying users of known vulnerabilities.

Vulnerability handling should be done in a timely manner, based on prioritization. Upon identification, vulnerabilities should be evaluated in terms of risk, scope of affected products, availability of mitigations, and other factors, and should be prioritized based on that evaluation. Organizations should allocate resources to address identified vulnerabilities according to that prioritization.

“Accepting vulnerability notifications” can be done in various ways. For example, an organization can participate in threat sharing programs, review posted threat information, work directly with third parties or publish information on how to reach a security team’s defined point-of-contact.

With regard to a security team’s defined point-of-contact, a useful “default” is security@company.com, where company.com is the organization’s email domain. Many third parties will attempt to contact an organization through this path. Despite this default’s popularity, however, the organization should have a “landing page” for contact information and policy on handling vulnerabilities.²⁴

5.2.2 EoL/EoS Updates and Disclosure

Definition: A defined manufacturer policy covering the handling of any post end-of-life (EoL) or end-of-service (EoS) device vulnerabilities, if and how updates will be available, and what to do with the device at EoL/EoS.

Scope: The published manufacturer policy on end of life and end of service.

Discussion: This capability must be considered carefully within the organization. It is tied to vulnerability handling, the product lifecycle, terms of service and more.

5.2.3 Device Intent Documentation

Definition: An explanation of the device’s as-designed network usage that is made available by the manufacturer publicly, in product documentation, or other means for device users.

Scope: Device use of network resources including communication with other devices; use of internet resources (including web sites); and with what protocols or services (e.g. UDP/TCP).

Discussion: The manufacturer or other responsible organization publishes, in a place readily accessible to device owners and operators, a summary of what behavior to expect from the device. An IoT device that is not intended to be on social media, or to scan port usage on the local intranet, or to contact devices made by other manufacturers, should not do these things. However, it is not always clear to the human monitoring the network whether a particular behavior is anomalous or not. The documentation should clarify this point. The user (or device administrator) should be able to readily determine what this device is intended to communicate with in terms of other devices; internet resources (including web sites); and with what protocols or services (e.g. UDP/TCP) as per the Scope.

It must also be noted that many device owners will not choose to use this information or may not have the training or experience necessary to use this information. However, it is important that the information be made available to those who can use it.



05 | Annex A: Regarding Future Secure Capabilities — Phase in Over Time

The following items are considered significant enough that they should be baseline capabilities. However, for various reasons they cannot be considered baseline at this time. The expectation is that they will become baseline and developers should carefully consider the capabilities in their planning.

A.1 Device Intent Signaling

Status: Baseline Capability to Phase In over Time

Definition: Means for the device to provide information to routers or firewalls upstream what kind of traffic the device was intended to produce.

Scope: This capability includes device-provided heuristics related to the device in normal operation (so that network analysis can be performed) and can include protocols such as Manufacturer Usage Descriptor (MUD)²⁵, OMA-DM²⁶ and TR-69²⁷ (the latter two being applicable in cases where the devices can be managed directly), security requirements including Open Connectivity Forum Security Profiles (Black, Blue and Purple), and proposals such as IoTSense.²⁸

Discussion: This capability will have a significant effect to reduce the scope and spread of botnets. There are test and implementation projects²⁹ under way to verify some of these technologies as well as discussions regarding appropriate use cases. Other voluntary consensus standards may be applied for similar capability.

It may also be helpful for the device to have its intent defined in a public way. As an example, a device manufacturer could simply have a published MUD file, regardless of whether the device supports emitting the URL or the network enforces the resulting intent. Knowing the device intent—even if it is simply via a plain text file as to what the device does—can help those seeking to correct anomalous behavior or reduce device threat surface.

Because the core technologies are documented but in testing, or available but not documented to this specific intent, this capability is considered one that should be phased in over time.

A.2 Device Network Onboarding

Status: Baseline Capability to Phase In over Time

Definition: Means to enable a network operator or device manager to cryptographically ensure that a device, when first attached to a network, is identified, authenticated and authorized.

Scope: Device onboarding is the process of authenticating the device, authorizing that device with credentials, and configuring it to be able to communicate within the security domain under question.

Discussion: From a security perspective, this process is one of the exchanges most fraught with peril. Correct identification of the device, and explicit, non-automated, approval from the network manager are both critical to the exchange.

Examples of Device Provisioning Protocols include the WiFi Alliance Device Provisioning Protocol (DPP).

Examples of Onboarding can be found in the OCF specifications.



06 | Annex B: Additional IoT Device Security Capabilities and Practices

This section identifies capabilities or practices that are not broadly applicable across the diverse IoT ecosystem and are therefore not intended to be part of the baseline. This does not mean that these capabilities are not important in the effort to secure the IoT ecosystem; it simply means that they are not suitable for a broad baseline.

B.1 Secure Development Lifecycle

Status: Not a Baseline Capability

Definition: Use of software assurance processes that consider security throughout the design, deployment, integration, and maintenance of software to reduce the risk of vulnerabilities and weaknesses.

Scope: A secure software development lifecycle (SDL) is a set of guidance and processes designed to ensure security considerations are addressed throughout the software's lifecycle. While specific elements of an SDL may vary, SDLs should include, at minimum, the following elements:

1. Processes to identify likely threats to the software and to map security controls and other mitigations to those threats in designing the software;
2. Processes to ensure that software code is written according to established voluntary consensus coding standards and avoids common weaknesses and vulnerabilities;
3. Processes to identify, vet, manage, and securely integrate third-party software components;
4. Processes to test and validate software security controls and capabilities; and
5. Processes to identify, manage, mitigate, and learn from new vulnerabilities, weaknesses, and advancements in best practices.

Discussion: Many IoT device manufacturers use software developed by third-parties; the intent of this item is to ensure that IoT device manufacturers obtain software components from software developers that have SDLs in place, and that manufacturers are able to obtain information from software developers about the nature and scope of their SDLs.

B.2 Hardware Rooted Security

Status: Not a Baseline Capability

Definition: The starting point of a chain of trustworthiness that includes a trusted execution environment with cryptographic functions, runtime execution tamper protection and an interface for the host process of the device.

Discussion: Hardware rooted security is important for certain aspects of security, but not all devices require it. Elements of hardware rooted security may include,

Secure or measured boot process. A secure boot process ensures that only the intended boot software is run. A measured boot process can signal an anomaly if other boot software is run because the boot process metrics will differ.

Protected or hardware cryptographic keys, which may be used to authenticate boot components or to uniquely identify the hardware device. A hardware-based key may also be used as a private key for software or application decryption. Multiple keys are typically required for the various purposes.

Trusted execution environment, which has access to reserved software and data for security purposes.

B.3 Time Distribution

Status: Not a Baseline Capability

Definition: Means to synchronize the device internal clock to wall clock time (e.g., UTC or “GPS time”).

Discussion: Time awareness could be an element of using good cryptographic methods; that is, some details of a secure device may need a reliable time indicator. For example, for logging of events, a known good timestamp is important.

However, implementing time distribution as a component of an overall cryptographic strategy or architecture implies that the time distribution protocol itself is a secure process. Time packets must be cryptographically signed for authentication to prevent man-in-the-middle attacks. If the time distribution protocol is used to manage key expiration, message spoofing can be a problem. See <http://www.cs.bu.edu/~goldbe/papers/NTPattack.pdf> for a discussion of security concerns for the NTP time distribution protocol.

B.4 System Resiliency

Status: Not a Baseline Capability



Definition: Ability to maintain service in the presence of certain kinds of faults.

Discussion: This capability is related to the principle of “least functionality”. This is a best practice. A device must be able to function post security attack (assuming the attack did not result in actual damage); however, the device may require software or firmware reinstatement or update.

The types of faults that may occur, and to which the device should be resilient include power outage or network outage; the latter includes specific network resource outage or unavailability such as the inability to contact a server or website. On resumed availability of power, network or resource, the device should be able to return to operation in a stable state.

B.5 Secure Toolchains

Status: Not a Baseline Capability

Definition: A set of programming tools to perform or automate large parts of the software development process that are designed and employed with security of the final product in mind.

Discussion: The toolchain is the suite of software tools used to develop, compile, build and maintain a software product, including the software or firmware embedded in an IoT device. Security-focused toolchains provide the capability to check if the implementation is following secure coding guidelines and to search for a subset of known Common Vulnerabilities and Exposures (CVEs) in the open source software.

Tools such as fuzzing, symbolic execution, sandboxing, static and dynamic analysis, and memory-safe languages can also be used to find and mitigate vulnerabilities.^{30, 31}

B.6 Software Transparency and Bill of Materials

Status: Not Baseline Capability

Definition: Ability to expose information about the software components used to build the device, including the components and their provenance.

Discussion: This capability is considered important, but currently the state of the art appears unready for Baseline status.

Software transparency refers to providing information regarding the sources of the devices software or firmware. The software bill of materials is an inventory of the device’s current internal software and firmware including versions and patches.³²

However, these features are still evolving and not fully available. For reference, note that NIST says in their *Considerations for a Core IoT Cybersecurity Capabilities Baseline* that software transparency “may offer utility [but] would be difficult to adequately verify and harder to implement”. They further note that the SBOM capability “is useful for update management but not necessary in all update mechanisms.”³³

There are technologies and products available, and requirements in certain markets, but these capabilities are not considered baseline.

B.7 Least Functionality

Status: Not a Baseline Capability

Definition: Ensuring that the device has only the necessary functions for operation.

Discussion: This is a good best practice but not a Baseline. Generally, it cannot be shown that a device exhibits least functionality, and at best a developer can only assert that they practice this during development.

B.8 Physical Access Control

Status: Not a Baseline Capability

Definition: Means to prevent a malicious actor from gaining undetected physical access to the unit, including tamper seals, conformal coating and physical locks.

Discussion: Physical access control is helpful to deter certain kinds of attacks. In many use cases, however, it is a consideration for the installer. Developers may consider tamper resistant coatings or tamper evidence seals.

B.9 Best Current Practices

Status: Not a Baseline Capability

Definition: Use of recommended industry practices and voluntary consensus standards.

Discussion: Best practices are important and should be considered by the developer and the organization.

However, this capability cannot be verified on the device.

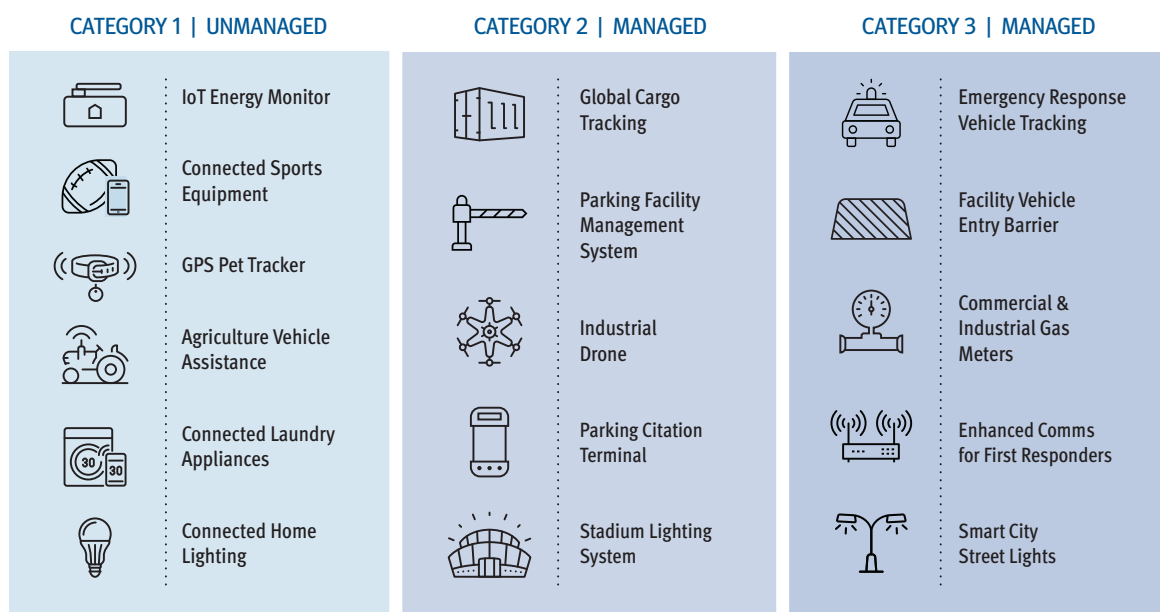
07 | Annex C: Discussion of Implementation and Complexity

The Baseline Capabilities described in the preceding sections can be implemented with various technologies, generally specified in voluntary consensus standards documents. Because there are options for each Capability, selection of the appropriate technology and appropriate implementation details (such as parameters or options) is critical. This section describes how to categorize device types to map to the appropriate levels of complexity in the implementation of the Baseline Capabilities.

As a common reference point, consider three categories of devices:

1. Category 1 devices are **low-complexity, unmanaged** devices (e.g., connected light bulb, connected appliances);
2. Category 2 devices are **medium-complexity, managed** devices (e.g., Industrial Drone, Global Cargo Tracking System);
3. Category 3 devices are **high-complexity, managed** devices (e.g., connected gas meter, connected city street lights).

FIGURE 1: USE CASE EXAMPLES



While it is critical that each of these types of devices is secure, they each require different security features because they have different degrees of complexity and manageability, and different use cases, which all result in different risk profiles. What is appropriate in an emergency response vehicle tracking system may not be appropriate in a GPS pet tracker.

Similarly, while some Baseline implementation methods or technologies are appropriate for Category 2 and Category 3 devices, they may be inapplicable to Category 1 devices. This does not mean that these features are not important in the effort to secure the IoT ecosystem; it simply means that they are not suitable for a broad Baseline. With this flexibility, the Baseline is appropriate for and applicable to all new IoT devices, even those at the lowest end of the complexity, sophistication, and manageability spectrum—like a connected light bulb.

By taking this approach—creating a Baseline for all new IoT devices, as opposed to creating a Baseline that is only relevant for complex or managed devices—the goal is to help to lift cybersecurity for the entire Internet of Things, including the network to which all of these devices connect.

Use Case Examples Where Complex Capabilities Are Not Appropriate

Encryption of all data at rest is a key example of a requirement that is not well-suited to all new IoT devices. In many instances, the data that would be protected is not sensitive or is less likely to be susceptible to misuse or danger if improperly accessed so that the tradeoffs for encrypting the data—including increased impacts on processor and decreased battery life—are high. As such, setting a baseline that expects all data at rest to always be encrypted in all IoT devices is fundamentally at odds with pragmatic risk management.

A few examples illustrate this:

- ▶ Adding encryption-at-rest to a GPS dog collar that has Wi-Fi or LTE connectivity will severely impact the battery life in that device. This tradeoff might make sense in some use cases, but it does not make sense with a low-complexity dog collar, which provides limited useful data.
- ▶ For IoT elements like low complexity sensors, where data is temporal or ephemeral, requirements to encrypt data at rest are unnecessary and burdensome.
- ▶ Connected footballs, soccer balls, and golf clubs may give users performance data from gyros embedded in them; however, on balance this is not the type of data that is likely to need to be encrypted at rest, and in fact, encrypting the data that is stored by these devices may degrade the utility of the device itself due to processor load.

Similar arguments can be made that other complex capabilities should not be treated as universal baselines, even if they may be desirable in many (or even most) use cases.

Many of these complex features are appropriate for other types of devices, namely Category 2 and Category 3 devices. The key is that beyond an agreed and basic universal baseline, there is not a one-size-fits-all solution: features to implement these capabilities should be determined based on the device's risk profile, which is informed by the device's complexity, sophistication, manageability, and general use case. For example,



- ▶ The risk profiles of general cargo tracking devices at sea, on rail, or on the road may call for specific security features, whereas a tracking device that monitors emergency response vehicles may call for other security features.
- ▶ The same logic would apply to a small stadium lighting system that needs to be connected to a school and a connected city lighting system that can dynamically adjust to multiple conditions depending on 911 calls, the presence of people in an area, gunshot detection technology, etc.
- ▶ A parking facility system may need added security because it combines entrance and exit with a point of sale terminal, while a high security entrance to a prison, an energy plant, or a military base would need even further enhanced security systems and added cybersecurity controls.

NIST has acknowledged that “[b]ecause IoT devices and their uses and needs are so varied, few recommendations can be made that apply to all IoT devices.” NISTIR 8228 DRAFT, <https://nvlpubs.nist.gov/nistpubs/ir/2018/NIST.IR.8228-draft.pdf>.

Because of this, a Baseline needs to be truly focused on those few Capabilities that are really universally applicable. Further, the implementation of this Baseline is subject to risk/complexity/management tradeoffs based on the risk, complexity and managed/unmanaged nature of the device type and application.

08 | Annex D: Informative References

The work of the C2 Consensus organizations draws on recommendations by these groups and others. The following references are informative.

- ▶ [IABG] Council to Secure the Digital Economy (CSDE), “*International Anti-Botnet Guide*”, November 2018, <https://securingdigitaleconomy.org/wp-content/uploads/2018/11/CSDE-Anti-Botnet-Report-final.pdf>
- ▶ [CTIA IoT CC] CTIA, *Cybersecurity Certification Test Plan for IoT Devices*, October 2018, https://api.ctia.org/wp-content/uploads/2018/10/CTIA-IoT-Cybersecurity-Certification-Test-Plan-V1_O_1.pdf
- ▶ [ETSI] European Telecommunications Standards Institute (ETSI), *TS 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
- ▶ [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>
- ▶ [Security Pledge] Internet of Secure Things (IoXT), *The IoXT Security Pledge*, <https://www.ioxtalliance.org/s/ioXt-SecurityPledge-booklet-final.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/>
- ▶ National Institute of Standards and Technology (NIST, United States Department of Commerce), *Considerations for a Core IoT Cybersecurity Capabilities Baseline*, February 2019, https://www.nist.gov/sites/default/files/documents/2019/02/01/final_core_iot_cybersecurity_capabilities_baseline_considerations.pdf
- ▶ [NISTIR 8259] National Institute of Standards and Technology (NIST, United States Department of Commerce), NISTIR 8259 (Draft), *Core Cybersecurity Feature Baseline for Securable IoT Devices: A Starting Point for IoT Device Manufacturers*, July 2019, <https://csrc.nist.gov/publications/detail/nistir/8259/draft>
- ▶ [OCF Security Specification ISO/IEC 30118-2:2018] Open Connectivity Foundation (OCF), *OCF Security Specification 2.0.1*, February 2019, https://openconnectivity.org/specs/OCF_Security_Specification_v2.0.1.pdf
- ▶ [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>
- ▶ [UL] UL, UL MCV 1376 — Security Capabilities Verified, <https://shopulstandards.com/ProductDetail.aspx?UniqueKey=35953>
- ▶ World Wide Web Coalition (W3C), *WoT Security Best Practices*, retrieved May 2019, <https://github.com/w3c/wot-security-best-practices>
- ▶ World Wide Web Coalition (W3C), *WoT Security Testing Plan*, retrieved May 2019, <https://github.com/w3c/wot-security-testing-plan>

09 | Annex E: Mapping to CSDE International Anti-Botnet Guide

| CATEGORY | SUB-CATEGORY | MAPS TO |
|---------------------------------------|---|--|
| Secure Device Capabilities - Baseline | Device Identifiers | [IABG] 5.C.3: Where possible, the device should support network asset management by enabling the ability to identify and audit the device logically and physically and with proper access control. |
| Secure Device Capabilities - Baseline | Secured Access | [IABG] 5.C.1.b.2: Unique "admin" credentials per device or a first-boot requirement to change passwords Rate-limiting techniques to prevent brute-force password guessing Securing or disabling developer-level ports and services prior to product shipment Removing unused or insecure local and remote administrative services such as telnet. Multi-factor authentication user access control should be supported. |
| Secure Device Capabilities - Baseline | Data In Transit Is Protected | [IABG] 5.C.1.b.1: Data communications should be encrypted. Regardless of whatever protocols are in use, if authentication is available, it should be used. In general, the security mechanisms available in whatever system is used should be employed. |
| Secure Device Capabilities - Baseline | Data At Rest Is Protected | [IABG] 5.C.1.b.1: Sensitive data should be stored encrypted. In general, the security mechanisms available in whatever system is used should be employed. |
| Secure Device Capabilities - Baseline | Industry Accepted Protocols are Used for Communications | [IABG] 5.C.1.b.1: The latest versions of protocols and security mechanisms should be used. Secure memory can be used in lieu of encryption for stored information. Encryption key methods comporting with NIST FIPS 140-2 or ISO/IEC 24759 should be used. |
| Secure Device Capabilities - Baseline | Data Validation | [IABG] 5.C.1.b.4: Any input received from outside the system must be managed so that an outside adversary cannot take advantage of unintended consequences. Input should be validated for length, character type, and acceptable values or ranges; see also whitelist filtering. Output from one subsystem to another or to another site should also be filtered; see "character canonicalization." |
| Secure Device Capabilities - Baseline | Event Logging | |
| Secure Device Capabilities - Baseline | Cryptography | [IABG] 5.C.1.b.1: Cryptographic techniques used should avoid deprecated methods. [IABG] 5.C.1.b.5: Cryptographic methods are required to ensure data integrity and confidentiality, rights authentication and non-repudiation of requests. This cryptography should be chosen to match the assessed risk but should use open, peer-reviewed methods and algorithms. Where feasible, cryptographic methods are updateable. ([IABG] Advanced section: Strong, proven, updateable cryptography using open, peer-reviewed methods and algorithms. Ensure cryptography has the ability to support post-quantum resistant key lengths for symmetric encryption.) |

| CATEGORY | SUB-CATEGORY | MAPS TO |
|---|---|--|
| Secure Device Capabilities - Baseline | Patchability | [IABG] 5.C.3: May provide notice to the consumer about security support policy and how the device is supported with updates during and what to expect after the support period. ([IABG] Advanced section: 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 | [IABG] 5.C.3: After the support period, consumers should have the ability to and be informed about how to “decommission” the device. Decommissioning should allow a consumer to return the product to factory defaults and remove any Personally Identifiable Information (PII). This capability covers a variety of scenarios such as the sale, abandonment, or recycling of the product, including selling a property with IoT devices installed. |
| Product Lifecycle Management | Vulnerability Submission and Handling Process | [IABG] 5.C.3: Providers should create a security vulnerability policy and process to identify, mitigate, and where appropriate, disclose known security vulnerabilities in their products. |
| Product Lifecycle Management | EoL/EoS Updates and Disclosure | [IABG] 5.C.3: May provide notice to the consumer about security support policy and how the device is supported with updates during and what to expect after the support period. |
| Produce Lifecycle Management | Device Intent Documentation | |
| Secure Capabilities - Phase In Over Time | Device Intent Signaling | [IABG] Multi-factor authentication user access control should be supported. IETF Manufacturer Usage Descriptor (MUD) may be supported; IEEE 802.1AR and the Device Identifier Composition Engine (DICE) should be considered to improve the security of the IoT device and its MUD components. |
| Secure Capabilities - Phase In Over Time | Device Network Onboarding | |
| Additional IoT Device Security Capabilities and Practices | Secure Development Lifecycle | [IABG] 5.C.1.a.1: A responsible company may have the secure development lifecycle (SDL) process. In the SDL process, each development phase has security activities that can be done manually or automatically. ([IABG] Advanced section: After establishing a secure development lifecycle process, the advanced company is measuring and growing process capabilities.) |
| Additional IoT Device Security Capabilities and Practices | Hardware Rooted Security | [IABG] 5.C.2.a.1: Consider how hardware-rooted security fits into the secure development lifecycles of current and future products. ([IABG] Advanced section: Hardware-rooted security is utilized where technically feasible.) |
| Additional IoT Device Security Capabilities and Practices | Time Distribution | |
| Additional IoT Device Security Capabilities and Practices | System Resiliency | |



| CATEGORY | SUB-CATEGORY | MAPS TO |
|---|---|--|
| Additional IoT Device Security Capabilities and Practices | Secure Toolchains | [IABG] 5.C.4: A responsible company may have 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) in the open source software. ([IABG] Advanced section: Tools such as fuzzing, symbolic execution, sandboxing, static and dynamic analysis, and memory-safe languages are used to find and mitigate vulnerabilities.) |
| 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 | |

10 | Annex F: Mapping to CTIA IoT Device Cybersecurity Certification

| CATEGORY | SUB-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 | <p>[CTIA IoT CC] 3.2: Password Management Test - Unique Default Password for each device</p> <p>Password Change required upon first login</p> <p>Password is of sufficient complexity and length</p> <p>[CTIA IoT CC] 3.3: Authentication Test - Authentication required to modify device settings</p> <p>[CTIA IoT CC] 3.4: Access Controls - Role Based Access Controls</p> <p>[CTIA IoT CC] 4.2: Password Management Test - Idle logout</p> <p>Password Integration with Enterprise Management System</p> <p>[CTIA IoT CC] 4.3: Access Control - Integrated password with Enterprise Management System</p> <p>[CTIA IoT CC] 4.9: Multi-factor Authentication - Multi-Factor Authentication is supported</p> <p>[CTIA IoT CC] 5.17 Designed In Feature - All Network Communications except those minimally required to function are disabled by default</p> |
| Secure Device Capabilities - Baseline | Data In Transit Is Protected | <p>[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.</p> <p>[CTIA IoT CC] 5.15 - Encryption of Data at Rest - Required support for encryption of data at rest at minimal 128-bit AES</p> |
| Secure Device Capabilities - Baseline | Data At Rest Is Protected | |
| Secure Device Capabilities - Baseline | Industry Accepted Protocols are Used for Communications | <p>[CTIA IoT CC] - CTIA recommends common peer reviewed industry standards</p> <p>Encryption in transit supports IPSEC, SSH, TLS and DTLS at the 128-bit AES support</p> <p>Encryption at Rest support ts minimal 128-bit AES support</p> <p>Digital Signature Generation and Validation support RSA or ECDSA algorithms for X.509 certificates in P7S formats</p> |
| Secure Device Capabilities - Baseline | Data Validation | <p>[CTIA IoT CC] 3.2 - validates inputs for password</p> <p>[CTIA IoT CC] 3.5/3.6 - validates patches and upgrades</p> <p>[CTIA IoT CC] 5.13 - validates digital certificates</p> <p>[CTIA IoT CC] 5.17 - validates network services minimally required</p> |
| 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 | SUB-CATEGORY | MAPS TO |
|---|---|--|
| Secure Device Capabilities - Baseline | Cryptography | <p>[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</p> <p>[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</p> <p>[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.</p> |
| 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 | <p>[CTIA IoT CC] 4.11 Secure Boot may be accomplished with the use of a hardware root of security such as a TPM module</p> <p>[CTIA IoT CC] 5.14 Digital Signature Generation and Validation may have a hardware root of trust module to support this functionality</p> |
| Additional IoT Device Security Capabilities and Practices | Time Distribution | |
| Additional IoT Device Security Capabilities and Practices | System Resiliency | |

| CATEGORY | SUB-CATEGORY | MAPS TO |
|---|---|--|
| 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 | |

11 | Annex G: Mapping to IoTopia Specifications

| CATEGORY | SUB-CATEGORY | MAPS TO |
|---------------------------------------|---|---|
| Secure Device Capabilities - Baseline | Device Identifiers | <p><i>Certificate based authentication. Onboarding requires a voucher with dev ID. MUD URL imbedded in device by manufacturer.</i></p> <ul style="list-style-type: none"> 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 | <p>“Device must utilize secure standard protocols and security mechanisms to provide multi-factor authentication for remote and local (physical) access to device</p> <ul style="list-style-type: none"> 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” <p>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.</p> |
| Secure Device Capabilities - Baseline | Data In Transit Is Protected | <p><i>Secure boot, secure data storage, measured boot, voucher storage, key storage, crypto support, crypto upgrade potential</i></p> <p>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.</p> |
| Secure Device Capabilities - Baseline | Data At Rest Is Protected | <p><i>Device manufacturer should provide Heuristics related to the device in normal operation so that network analysis can be performed</i></p> |
| Secure Device Capabilities - Baseline | Industry Accepted Protocols are Used for Communications | <p><i>Device must support industry standard protocols internally and for data transmission egress</i></p> <p>An Endpoint that communicates via IEEE 802 networking must support [RFC7030], Section 3 on TLS Layer, for certificate management of secure transport.</p> <p>Endpoints must measure secure boot: Secure boot is a ‘security mechanism’ and measured boot is the monitoring required</p> <p>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</p> |
| Secure Device Capabilities - Baseline | Data Validation | |

| CATEGORY | SUB-CATEGORY | MAPS TO |
|--|---|--|
| 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 |
| Secure Device Capabilities - Baseline | Cryptography | <p>a. Cryptography: The endpoint MUST support the SHA-256 hash algorithm</p> <p>b. The endpoint must support for Elliptic Curve Cryptography (ECC) described in [RFC6090] and [IEEE802.1AR] for use as LDevIDs</p> <p>c. An Endpoint must support either 2048-bit RSA certificates or ECC certificates as described in [RFC6090] and [IEEE802.1AR] for iDevIDs</p> <p>d. TLS Cipher Suite Support: Endpoints must minimally support the TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 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.</p> <p>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.</p> |
| Secure Device Capabilities - Baseline | Patchability | <p><i>Device and manufacturer support secure SW/FW/HW updates throughout device lifecycle</i></p> <p>a. Endpoints must have the ability to securely receive and apply a software and/or firmware update</p> <p>b. All updates must be signed by the manufacturer, and Endpoints must validate signatures prior to applying any updates</p> <p>c. Endpoints that implement via IEEE 802 networking must support installation of at least one local certificate (LDevIDs) and associated private keying material</p> |
| Secure Device Capabilities - Baseline | Reprovisioning | <i>Device must support secure, authorized access control for remote and physical connection to device</i> |
| Product Lifecycle Management | Vulnerability Submission and Handling Process | <p><i>Manufacturer must provide any known device vulnerabilities and a plan or process to mitigate such vulnerabilities</i></p> <p><i>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.</i></p> |
| Product Lifecycle Management | EoL/EoS Updates and Disclosure | <i>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)</i> |
| Produce Lifecycle Management | Device Intent Documentation | |
| Secure Capabilities - Phase In Over Time | Device Intent Signaling | <p>Manufacturer must provide a file server that distributes Manufacturer Usage Description (MUD) files in accordance with MUD RFC</p> <p>a. The MUD-URL mustT 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 mustT also present an unsecured statement of the MUD-URL via LLDP or DHCP</p> <p>b. Endpoints must only run applications or services whose TCP or UDP ports are described in the MUD profile</p> |

| CATEGORY | SUB-CATEGORY | MAPS TO |
|---|---|--|
| Secure Capabilities - Phase In Over Time | Device Network Onboarding | Device must support MUD URIs to provide the network with information to microsegment/set ACL's. In addition the device should support an automated onboarding capability such as BRSKI |
| Additional IoT Device Security Capabilities and Practices | Secure Development Lifecycle | <p>Vendors must have a written SDL process in place that includes the following elements at a minimum:</p> <ul style="list-style-type: none"> • 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. <p>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.</p> |
| Additional IoT Device Security Capabilities and Practices | Hardware Rooted Security | <p>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.</p> <p>Endpoints must store private keying material and certificates in tamperproof storage</p> |
| Additional IoT Device Security Capabilities and Practices | Time Distribution | <p>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</p> <p>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.</p> |
| 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 | <p><i>Device should provide mitigation options including device shut-down in the event of a security attack/breach</i></p> <p>a. Network elements must support limited network access for endpoints that do not support 802.1X</p> <p>b. Upon detecting a threat, anNetwork must isolate infected devices based on local policy and report the action to the network administrator.</p> |

| CATEGORY | SUB-CATEGORY | MAPS TO |
|---|-------------------------|--|
| Additional IoT Device Security Capabilities and Practices | Physical Access Control | <p><i>Device should be able to block un-authorized physical access. For direct connection to a device there must be a secure/authorization process</i></p> <p>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.</p> |
| Additional IoT Device Security Capabilities and Practices | Best Current Practices | |

12 | Annex H: Mapping to IoXT Pledge

| CATEGORY | SUB-CATEGORY | MAPS TO |
|---------------------------------------|---|--|
| Secure Device Capabilities - Baseline | Device Identifiers | |
| Secure Device Capabilities - Baseline | Secured Access | <p>[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.</p> <p>[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.</p> |
| Secure Device Capabilities - Baseline | Data In Transit Is Protected | <p>[Security Pledge] 2. Secured Interfaces In all cases, any external communication interfaces shall be secured. All sensitive interfaces shall be encrypted and authenticated.</p> |
| Secure Device Capabilities - Baseline | Data At Rest Is Protected | |
| Secure Device Capabilities - Baseline | Industry Accepted Protocols are Used for Communications | <p>[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.</p> |
| Secure Device Capabilities - Baseline | Data Validation | <p>[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.</p> <p>[Security Pledge] 2: Secured Interfaces All sensitive interfaces shall be encrypted and authenticated.</p> |
| Secure Device Capabilities - Baseline | Event Logging | |
| Secure Device Capabilities - Baseline | Cryptography | <p>[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.</p> |

| CATEGORY | SUB-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. |
| Produce 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 | |
| Additional IoT Device Security Capabilities and Practices | Secure Toolchains | |



| CATEGORY | SUB-CATEGORY | MAPS TO |
|---|---|--|
| 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 | |

13 | Annex I: 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 | SUB-CATEGORY | MAPS TO |
|---------------------------------------|---|--|
| Secure Device Capabilities - Baseline | Device Identifiers | [OCF Security Specification ISO/IEC 30118-2:2018] Clause 7.1.1. The unique identifier for the device is either sent in the certificate the device sends when establishing communication on the network, or bound to a pre-shared key. |
| Secure Device Capabilities - Baseline | Secured Access | [OCF Security Specification ISO/IEC 30118-2:2018] Clauses: 5,6,7: 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 13.3.1: Stored Credentials used to authenticate server to clients. Note: OCF does not specify physical access controls. |
| Secure Device Capabilities - Baseline | Data In Transit Is Protected | [OCF Security Specification ISO/IEC 30118-2:2018] Clause 11.2.1: Devices must support TLS/DTLS version 1.2 or greater for all unicast sessions. |
| Secure Device Capabilities - Baseline | Data At Rest Is Protected | [OCF Security Specification ISO/IEC 30118-2:2018] Clause 14.2.2: Secure storage for credentials is strongly recommended. [OCF Vendor Attestation Document]: Certification applicant has taken appropriate measures to protect Sensitive Data as defined in OCF Security Specification ISO/IEC 30118-2:2018 Clause 14.2.2 |
| 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 11.2.1: Devices must support CoAP, and CoAP over DTLS. [OCF Security Specification ISO/IEC 30118-2:2018] Clause 11.3: 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. |
| Secure Device Capabilities - Baseline | Event Logging | Future work for OCF |

| CATEGORY | SUB-CATEGORY | MAPS TO |
|--|---|---|
| Secure Device Capabilities - Baseline | Cryptography | [OCF Security Specification ISO/IEC 30118-2:2018] Clause 11.3.1: 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 ISO/IEC 30118-2:2018] Clause 14.5.3: Process where device validates the software version against a trusted source. [OCF Security Specification ISO/IEC 30118-2:2018] 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 8.2 Defines how resources on the device are returned to the manufacturer's default values. |
| Product Lifecycle Management — Baseline | 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 — Baseline | EoL/EoS Updates and Disclosure | [OCF] Updatable Certified Product List: Website. https://openconnectivity.org/certified-products manufacturers should notify OCF that device is EoL. |
| Produce Lifecycle Management - Baseline | Device Intent Documentation | [OCF Security Specification ISO/IEC 30118-2:2018] 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 systeminfo (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 ISO/IEC 30118-2:2018] 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 Security Specification ISO/IEC 30118-2:2018] Clause 3.1.31 Device Configuration Resource (DCR): 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 5.3 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 | SUB-CATEGORY | MAPS TO |
|---|---|--|
| Additional IoT Device Security Capabilities and Practices | Secure Development Lifecycle | [OCF Security Specification ISO/IEC 30118-2:2018] 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 | <p>[OCF Security Specification ISO/IEC 30118-2:2018] 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.</p> <p>[OCF Security Specification ISO/IEC 30118-2:2018] Clause 14.2.2.2: Hardware Secure storage is recommended for symmetric and asymmetric keys, access credentials and personal private data.</p> <p>[OCF Security Specification ISO/IEC 30118-2:2018] Clause 14.2.7: Defines levels of Hardware Tamper Protection for cryptographic module.</p> |
| Additional IoT Device Security Capabilities and Practices | Time Distribution | [OCF Security Specification ISO/IEC 30118-2:2018] 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]: 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 ISO/IEC 30118-2:2018] 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 | <p>[OCF Security Specification ISO/IEC 30118-2:2018] Clause 14.2.7: Defines levels of Hardware Tamper Protection for cryptographic module.</p> <p>[OCF Security Specification ISO/IEC 30118-2:2018] Clause 14.2.2.4: Additional Security Guidelines and Best Practice</p> |
| Additional IoT Device Security Capabilities and Practices | Best Current Practices | [OCF Security Specification ISO/IEC 30118-2:2018] 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. |

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

| CATEGORY | SUB-CATEGORY | MAPS TO |
|---|---|---|
| Secure Device Capabilities - Baseline | Device Identifiers | |
| Secure Device Capabilities - Baseline | Secured Access | |
| Secure Device Capabilities - Baseline | Data In Transit Is Protected | |
| 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 | |
| Secure Device Capabilities - Baseline | Event Logging | |
| Secure Device Capabilities - Baseline | Cryptography | See https://github.com/w3c/wot-security-best-practices |
| 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 | |
| Produce 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 | |

| CATEGORY | SUB-CATEGORY | MAPS TO |
|---|---|---|
| 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 | see https://github.com/w3c/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 | See https://github.com/w3c/wot-security-best-practice |

15 | Annex K: Mapping to EU Agency for Cybersecurity Baseline Security Recommendations for IoT

This section maps this group's recommendations³⁴ to the C2 Consensus. Note that the EU Agency for Cybersecurity was previously known as ENISA.

| CATEGORY | SUB-CATEGORY | MAPS TO |
|---------------------------------------|--------------------|--|
| Secure Device Capabilities - Baseline | Device Identifiers | [ENISA] (Annex A): GP-PS-10: Establish and maintain asset management procedures and configuration controls for key network and information systems, to identify and authenticate of the assets involved in the IoT Service (i.e. Gateways, Endpoint devices, home network, roaming networks, service platforms, etc.). |
| Secure Device Capabilities - Baseline | Secured Access | <p>[ENISA] (Annex A): GP-TM-09: Establish hard to crack device individual default passwords. Usernames and passwords for IoT devices supplied by the manufacturer are often never changed by the user and are easily cracked, and a hard to crack default password is still a weakness if it is used for more than one device.</p> <p>[ENISA] (Annex A): GP-TM-21: Design the authentication and authorisation schemes (unique per device) based on the system-level threat models. Devices should include mechanisms to reliably authenticate their backend services and supporting applications.</p> <p>[ENISA] (Annex A): GP-TM-22: Ensure default passwords and even default usernames are changed during the initial setup, and that weak, null or blank passwords are not allowed.</p> <p>[ENISA] (Annex A): GP-TM-23: Authentication mechanisms must use strong passwords or personal identification numbers (PINs), and should consider using two-factor authentication (2FA) or multi-factor authentication (MFA) like Smartphones, Biometrics, etc., and certificates.</p> <p>[ENISA] (Annex A): GP-TM-24: Authentication credentials including but not limited to user passwords shall be salted, hashed and/or encrypted</p> <p>[ENISA] (Annex A): GP-TM-25: Protect against 'brute force' and/or other abusive login attempts (such as automated login bots, etc.) by locking or disabling user and device support account(s) after a reasonable number of invalid log in attempts, or by making the user wait a certain amount of time to login again after a failed attempt. This protection should also consider keys stored in devices.</p> <p>[ENISA] (Annex A): GP-TM-26: Ensure password recovery or reset mechanism is robust and does not supply an attacker with information indicating a valid account. The same applies to key update and recovery mechanisms.</p> <p>[ENISA] (Annex A): GP-TM-27: Limit permissions of the allowed actions for a given system (e.g., the information owner or the database administrator determines who can update a shared file accessed by a group of online users). Implement fine-grained authorisation mechanisms - such as Attribute-Based Access Control (ABAC) or Role-Based Access Control (RBAC)- for executing privileged actions, access to files and directories, applications, etc. Use the Principle of least privilege (POLP): applications must operate at the lowest privilege level possible.</p> <p>[ENISA] (Annex A): GP-TM-43: IoT devices should be restrictive rather than permissive in communicating: When possible, devices should not be reachable via inbound connections by default. IoT devices should not rely on the network firewall alone to restrict communication, as some communication between devices within the home may not traverse the firewall.</p> |

| CATEGORY | SUB-CATEGORY | MAPS TO |
|---------------------------------------|---|---|
| Secure Device Capabilities - Baseline | Data In Transit Is Protected | <p>[ENISA] (Annex A): GP-PS-10: Establish and maintain asset management procedures and configuration controls for key network and information systems, to identify and authenticate of the assets involved in the IoT Service (i.e. Gateways, Endpoint devices, home network, roaming networks, service platforms, etc.).</p> <p>[ENISA] (Annex A): GP-TM-34: Ensure a proper and effective use of cryptography to protect the confidentiality, authenticity and/or integrity of data and information (including control messages), in transit and in rest. Ensure the proper selection of standard and strong encryption algorithms and strong keys, and disable insecure protocols. Verify the robustness of the implementation.</p> <p>[ENISA] (Annex A): GP-TM-38: Guarantee the different security aspects -confidentiality (privacy), integrity, availability and authenticity- of the information in transit on the networks or stored in the IoT application or in the Cloud, using data encryption methods to minimise network threats such as replay, interception, packet sniffing, wiretapping, or eavesdropping.</p> <p>[ENISA] (Annex A): GP-TM-39: Ensure that communication security is provided using state-of-the-art, standardised security protocols, such as TLS for encryption.</p> <p>[ENISA] (Annex A): GP-TM-40: Ensure credentials are not exposed in internal or external network traffic</p> <p>[ENISA] (Annex A): GP-TM-41: Guarantee data authenticity to enable trustable exchanges (from data emission to data reception - both ways). Data is often stored, cached, and processed by several nodes; not just sent from point A to point B. For these reasons, data should always be signed whenever and wherever the data is captured and stored.</p> <p>[ENISA] (Annex A): GP-TM-42: Do not trust data received and always verify any interconnections. Discover, identify and verify/authenticate the devices connected to the network before trust can be established, and preserve their integrity for trustable solutions and services. For example, a device measures its own integrity as part of boot, but does not validate those measurements - when the device applies to join a network, part of joining involves sending an integrity report for remote validation. If validation fails, the end point is diverted to a remediation network for action.</p> <p>[ENISA] (Annex A): GP-TM-43: IoT devices should be restrictive rather than permissive in communicating: When possible, devices should not be reachable via inbound connections by default. IoT devices should not rely on the network firewall alone to restrict communication, as some communication between devices within the home may not traverse the firewall.</p> |
| Secure Device Capabilities - Baseline | Data At Rest Is Protected | |
| Secure Device Capabilities - Baseline | Industry Accepted Protocols are Used for Communications | <p>[ENISA] (Annex A): GP-OP-04: Use proven solutions, i.e. well known communications protocols and cryptographic algorithms, recognized by the scientific community, etc. Certain proprietary solutions, such as custom cryptographic algorithms, should be avoided. Purely proprietary approaches and standards limit interoperability and can severely hamper the potential of the Digital Single Market. Common open standards will help users access new innovative services, especially for SMEs, the public sector and the scientific community. In particular, the portability of applications and data between different providers is essential to avoid lock-in.</p> |
| Secure Device Capabilities - Baseline | Data Validation | <p>[ENISA] 4.3.13 Secure input and output handling</p> <p>GP-TM-54: Data input validation (ensuring that data is safe prior to use) and output filtering.</p> |

| CATEGORY | SUB-CATEGORY | MAPS TO |
|---------------------------------------|----------------|---|
| Secure Device Capabilities - Baseline | Event Logging | [ENISA] (Annex A): GP-TM-55: Implement a logging system that records events relating to user authentication, management of accounts and access rights, modifications to security rules, and the functioning of the system. The logs must also be preserved on durable storage and retrievable via an authenticated connection. |
| Secure Device Capabilities - Baseline | Cryptography | <p>[ENISA] (Annex A): GP-TM-35: Cryptographic keys must be securely managed. Encryption is only as robust as the ability for any encryption based system to keep the encryption key hidden. Cryptographic key management includes key generation, distribution, storage, and maintenance.</p> <p>[ENISA] (Annex A): GP-TM-36: Build devices to be compatible with lightweight encryption and security techniques (including entities secure identification, secure configuration, etc.) that can, on the one hand, be usable on resource-constrained devices, and, on the other hand, be scalable so to minimise the management effort and maximise their usability.</p> <p>[ENISA] (Annex A): GP-TM-37: Support scalable key management schemes. It has to be considered that tiny sensor nodes cannot provide all security features because they have lots of system limitations. Thus, the sensed data carried over infrastructure networks may not have strong encryption or security protection.</p> |
| Secure Device Capabilities - Baseline | Patchability | <p>[ENISA] (Annex A): GP-TM-18: Ensure the device software/firmware, its configuration and its applications have the ability to update Over-The-Air (OTA), that the update server is secure, that the update file is transmitted via a secure connection, that it does not contain sensitive data (e.g. hardcoded credentials), and that it is signed by an authorised trust entity and encrypted using accepted encryption methods, and that the update package has its digital signature, signing certificate and signing certificate chain, verified by the device before the update process begins.</p> <p>[ENISA] (Annex A): GP-TM-19: Offer an automatic firmware update mechanism. Devices should be configured to check for the existence of firmware updates at frequent intervals. Automatic firmware updates should be enabled by default. A device may offer an option to disable automatic firmware updates and require authentication for it.</p> <p>[ENISA] (Annex A): GP-TM-20: Backward compatibility of firmware updates. Automatic firmware updates should not change network protocol interfaces in any way that is incompatible with previous versions. Updates and patches should not modify user-configured preferences, security, and/or privacy settings without user notification. Users should have the ability to approve, authorise or reject updates.</p> |
| Secure Device Capabilities - Baseline | Reprovisioning | <p>[ENISA] (Annex A): GP-OP-01: Develop an end-of-life strategy for IoT products. Security patches and updates will eventually be discontinued for some IoT devices. Therefore, developers should prepare and communicate a product sunset plan from the initial stages to ensure that manufacturers and consumers are aware of the risks posed to a device beyond its expected expiry date.</p> <p>[ENISA] (Annex A): GP-OP-02: Disclose the duration and end-of-life security and patch support (beyond product warranty). Such disclosures should be aligned to the expected lifespan of the device and communicated to the consumer prior to purchase.</p> |

| CATEGORY | SUB-CATEGORY | MAPS TO |
|---|---|--|
| Product Lifecycle Management | Vulnerability Submission and Handling Process | <p>[ENISA] (Annex A): GP-OP-03: Monitor the performance and patch known vulnerabilities up until the “end-of-support” period of a product’s lifecycle. Due to the limited life cycle of many IoT devices, critical, publicly known security or privacy bugs will pose a risk to consumers using outdated devices.</p> <p>[ENISA] (Annex A): GP-OP-05: Establish procedures for analysing and handling security incidents. For any incident there should be a response to:</p> <ul style="list-style-type: none"> a. confirm the nature and extent of the incident; b. take control of the situation; c. contain the incident; and d. communicate with stakeholders <p>Establish management procedures in order to ensure a quick, effective and orderly response to information security incidents.</p> <p>ENISA] (Annex A): GP-OP-06: Coordinated disclosure of vulnerabilities, including associated security practices to address identified vulnerabilities. A coordinated disclosure policy should involve developers, manufacturers, and service providers, and include information regarding any vulnerabilities reported to a computer security incident response team (CSIRT).</p> <p>ENISA] (Annex A): GP-OP-07: Participate in information sharing platforms to report vulnerabilities and receive timely and critical information about current cyber threats and vulnerabilities from public and private partners. Information sharing is a critical tool in ensuring stakeholders are aware of threats as they arise.</p> <p>ENISA] (Annex A): GP-OP-08: Create a publicly disclosed mechanism for vulnerability reports. Bug Bounty programs, for example, rely on crowdsourcing methods to identify vulnerabilities that companies’ own internal security teams may not catch.</p> |
| Produce 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 | <p>[ENISA] (Annex A): GP-TM-01: Employ a hardware-based immutable root of trust. The Hardware Root of Trust is a trusted hardware component which receives control at power-on. It then extends the chain of trust to other hardware, firmware, and software components. The Root of Trust should then be attestable by software agents running within and throughout the infrastructure.</p> <p>[ENISA] (Annex A): GP-TM-02: Use hardware that incorporates security features to strengthen the protection and integrity of the device — for example, specialised security chips/ coprocessors that integrate security at the transistor level, embedded in the processor, that provide: (see [ENISA] document for full list)</p> |



| CATEGORY | SUB-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 | |
| 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 | |

16 | Annex L: Mapping to ETSI 103 645

| CATEGORY | SUB-CATEGORY | MAPS TO |
|---------------------------------------|---|---|
| Secure Device Capabilities - Baseline | Device Identifiers | |
| Secure Device Capabilities - Baseline | Secured Access | <p>[ETSI] 4.1: No universal default passwords</p> <p>[ETSI] Provision 4.1-1 All IoT device passwords shall be unique and shall not be resettable to any universal factory default value.</p> <p>[ETSI] Provision 4.6-1 Unused software and network ports should be closed.</p> <p>[ETSI] Provision 4.6-2 Hardware should not unnecessarily expose access to attack (e.g. open serial access, ports or test points).</p> <p>[ETSI] Provision 4.6-3 Software services should not be available if they are not used.</p> |
| Secure Device Capabilities - Baseline | Data In Transit Is Protected | <p>[ETSI] 4.4 Securely store credentials and security-sensitive data</p> <p>[ETSI] 4.5 Communicate securely</p> <p>[ETSI] Provision 4.5-1 Security-sensitive data, including any remote management and control, should be encrypted in transit, with such encryption appropriate to the properties of the technology and usage.</p> <p>[ETSI] Provision 4.5-2 All keys should be managed securely.</p> |
| Secure Device Capabilities - Baseline | Data At Rest Is Protected | <p>[ETSI] Provision 4.10-1 If telemetry data is collected from IoT devices and services, such as usage and measurement data, it should be examined for security anomalies.</p> <p>[ETSI] Provision 4.10-2 If telemetry data is collected from IoT devices and services, the processing of personal data should be kept to a minimum and such data should be anonymized.</p> <p>[ETSI] Provision 4.10-3 If telemetry data is collected from IoT devices and services, consumers shall be provided with information on what telemetry data is collected and the reasons for this.</p> |
| Secure Device Capabilities - Baseline | Industry Accepted Protocols are Used for Communications | <p>[ETSI] 4.5 Communicate securely</p> <p>The use of open, peer-reviewed standards is strongly encouraged.</p> |
| Secure Device Capabilities - Baseline | Data Validation | <p>[ETSI] Provision 4.13-1 Data input via user interfaces and transferred via application programming interfaces (APIs) or between networks in services and devices shall be validated.</p> |
| Secure Device Capabilities - Baseline | Event Logging | <p>[ETSI] Provision 4.7-2 If an unauthorized change is detected to the software, the device should alert the consumer and/or administrator to an issue and should not connect to wider networks than those necessary to perform the alerting function.</p> |
| Secure Device Capabilities - Baseline | Cryptography | |

| CATEGORY | SUB-CATEGORY | MAPS TO |
|--|---|---|
| Secure Device Capabilities - Baseline | Patchability | <p>[ETSI] 4.3 Keep software updated</p> <p>Provision 4.3-1 All software components in consumer IoT devices should be securely updateable.</p> <p>Provision 4.3-2 The consumer should be informed by the appropriate entity, such as the manufacturer or service provider, that an update is required.</p> <p>Provision 4.3-3 When software components are updateable, updates shall be timely.</p> <p>Provision 4.3-4 When software components are updateable, an end-of-life policy shall be published for devices that explicitly states the minimum length of time for which a device will receive software updates and the reasons for the length of the support period. This policy shall be published in an accessible way that is clear and transparent to the consumer.</p> <p>Provision 4.3-5 When software components are updateable, the need for each update should be made clear to consumers and an update should be easy to implement.</p> <p>Provision 4.3-6 When software components are updateable, updates should, where possible, maintain the basic functioning of the device, which can be critical to remain available during an update.</p> <p>Provision 4.3-7 When software components are updateable, the provenance of software updates should be assured and security patches should be delivered over a secure channel.</p> <p>Provision 4.3-8 For constrained devices that cannot have their software updated, the product should be isolable and the hardware replaceable.</p> <p>Provision 4.3-9 For constrained devices that cannot have their software updated, the rationale for the absence of software updates, the period of hardware replacement support and an end-of-life policy should be published in an accessible way that is clear and transparent to the consumer.</p> |
| Secure Device Capabilities - Baseline | Reprovisioning | |
| Product Lifecycle Management | Vulnerability Submission and Handling Process | <p>[ETSI] 4.2: Implement a means to manage reports of vulnerabilities</p> <p>Provision 4.2-1: Companies that provide internet-connected devices and services shall provide a public point of contact as part of a vulnerability disclosure policy in order that security researchers and others are able to report issues.</p> <p>Provision 4.2-2 Disclosed vulnerabilities should be acted on in a timely manner.</p> <p>Provision 4.2-3 Companies should continually monitor for, identify and rectify security vulnerabilities within products and services they sell, produce, have produced and services they operate as part of the product security lifecycle.</p> |
| Product Lifecycle Management | EoL/EoS Updates and Disclosure | |
| Produce Lifecycle Management | Device Intent Documentation | |
| Secure Capabilities - Phase In Over Time | Device Intent Signaling | |

| CATEGORY | SUB-CATEGORY | MAPS TO |
|---|---|--|
| Secure Capabilities - Phase In Over Time | Device Network Onboarding | [ETSI] 4.12 Make installation and maintenance of devices easy [ETSI] Provision 4.12-1 Installation and maintenance of IoT devices should employ minimal steps and should follow security best practice on usability. Consumers should also be provided with guidance on how to securely set up their device. |
| Additional IoT Device Security Capabilities and Practices | Secure Development Lifecycle | |
| Additional IoT Device Security Capabilities and Practices | Hardware Rooted Security | Provision 4.7-1 Software on IoT devices should be verified using secure boot mechanisms, which require a hardware root of trust. |
| Additional IoT Device Security Capabilities and Practices | Time Distribution | |
| Additional IoT Device Security Capabilities and Practices | System Resiliency | [ETSI] Provision 4.9-1 Resilience should be built in to IoT devices and services where required by their usage or by other relying systems, taking into account the possibility of outages of data networks and power. [ETSI] Provision 4.9-2 As far as reasonably possible, IoT services should remain operating and locally functional in the case of a loss of network and should recover cleanly in the case of restoration of a loss of power. [ETSI] Provision 4.9-3 Devices should be able to return to a network in an expected, operational and stable state and in an orderly fashion, rather than in a massive-scale reconnect. |
| 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 | [ETSI] Provision 4.6-4 Code should be minimized to the functionality necessary for the service/device to operate. [ETSI] Provision 4.6-5 Software should run with least necessary privileges, taking account of both security and functionality. |
| 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 GSMA IoT Security Guidelines for Endpoint Ecosystems

| CATEGORY | SUB-CATEGORY | MAPS TO |
|---------------------------------------|---|---|
| Secure Device Capabilities - Baseline | Device Identifiers | |
| Secure Device Capabilities - Baseline | Secured Access | <p>[GSMA] 6.9 Endpoint Password Management</p> <p>Devices that incorporate user interfaces must be capable of managing passwords effectively. This requires several things</p> <ul style="list-style-type: none"> • Brute-force attack mitigation • Disabling the use of default or hardcoded passwords • Password best-practice enforcement • Disallowing display of user credentials on login interfaces • Enforcing thresholds and incremental delays for invalid password attempts <p>[GSMA] 6.12 Remote Endpoint Administration</p> <p>While not all Endpoints require remote administration, the ones that do must be architected in a way that ensures that third parties cannot abuse administrative credentials to compromise some (or all) of the Endpoints in the field. The appropriate solution will depend on the capabilities of the Endpoint</p> |
| Secure Device Capabilities - Baseline | Data In Transit Is Protected | <p>[GSMA] 6.14 Enforce Memory Protection</p> <p>Embedded systems are often designed with microcontrollers that are not capable of robust technology such as Memory Management Units (MMU) and Memory Protection Units (MPU)... implement memory protection with either an MPU or MMU.</p> <p>[GSMA] 6.15 Bootloading Outside of Internal ROM</p> <p>Consider using a CPU or MCU/MPU with an internal ROM or lock-capable NVRAM to store the bootloader. This will help to ensure that the platform can at least verify the first executable loaded and executed by the architecture, resulting in a more trustworthy device.</p> <p>[GSMA] 6.16 Locking Critical Sections of Memory</p> <p>Critical applications stored in executable regions of memory, such as first-stage bootloaders or Trusted Computing Bases, should be stored read-only</p> |
| Secure Device Capabilities - Baseline | Data At Rest Is Protected | |
| Secure Device Capabilities - Baseline | Industry Accepted Protocols are Used for Communications | <p>[GSMA] 6.19 Endpoint Communications Security</p> <p>This process is made far simpler through the use of existing and well analysed security protocols, such as, but not limited to:</p> <ul style="list-style-type: none"> • The latest approved TLS standard • The latest approved DTLS standard • SSH2 for authentication and key exchange • GBA for key generation and exchange • OAuth2 for authorization |

| CATEGORY | SUB-CATEGORY | MAPS TO |
|---|---|--|
| Secure Device Capabilities - Baseline | Data Validation | |
| Secure Device Capabilities - Baseline | Event Logging | <p>[GSMA] 6.13 Logging and Diagnostics</p> <p>In order to assess problems with Endpoint devices, the IoT Service Provider should constantly evaluate the behaviour of the Endpoint and determine whether the Endpoint is functioning within the set of approved behaviours. To accomplish this, three strategies should be used</p> <ul style="list-style-type: none"> • Anomaly detection • Endpoint logging • Endpoint 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 | |
| Secure Capabilities - Phase In Over Time | Device Intent Signaling | |
| Secure Capabilities - Phase In Over Time | Device Network Onboarding | <p>[GSMA] 6.8 Uniquely Provision Each Endpoint</p> <p>While personalization guarantees that each device is unique once it is manufactured, provisioning ensures that a unique device is activated, updated, and associated with a particular customer identity. The provisioning process helps separate devices that have been manufactured from devices that have been purchased and/or deployed in an IoT environment.</p> <p>[GSMA] 6.20 Authenticating an Endpoint Identity</p> <p>If each Endpoint carries a cryptographically unique identity, such as a unique serial number, the device must be able to prove that it truly represents that serial number.</p> |
| Additional IoT Device Security Capabilities and Practices | Secure Development Lifecycle | |



| CATEGORY | SUB-CATEGORY | MAPS TO |
|---|---|---|
| Additional IoT Device Security Capabilities and Practices | Hardware Rooted Security | <p>[GSMA] 6.1 Implement an Endpoint Trusted Computing Base</p> <p>The first step in securing any embedded system is the definition of the Trusted Computing Base (TCB). In the context of an Endpoint (or similar embedded devices), a TCB is a suite composed of hardware, software, and protocols that ensures the integrity of the Endpoint, performs mutual authentication with network peers, and manages communications and application security.</p> <p>[GSMA] 6.2 Utilize a Trust Anchor</p> <p>In order for an Endpoint to participate in an ecosystem, it must be able to verify the integrity of its own platform, and must be able to authenticate the identity of its peers. To do this, Endpoints require a trust anchor incorporated into a Trusted Computing Base.</p> <p>A trust anchor is a secure hardware element, either a separate physical chip, or a secure core inside a CPU, that is capable of securely storing and processing cryptographic secrets. A UICC or eUICC device is an example of a secure technology that can be used as a trust element to store authentication secrets.</p> <p>[GSMA] 6.3 Use a Tamper Resistant Trust Anchor</p> <p>[GSMA] 6.4 Define an API for Using the TCB</p> <p>[GSMA] 6.5 Defining an Organizational Root of Trust</p> <p>[GSMA] 6.6 Personalize Each Endpoint Device Prior to Fulfilment</p> <p>[GSMA] 6.7 Minimum Viable execution Platform (Application Roll-Back)</p> |
| 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 | |

18 | Annex N: Mapping to Draft NISTIR 8259

As of this writing, NISTIR 8259 is released in draft form for public comment. The mapping below shows likely places to explore the commonalities between the C2 Consensus baseline and the NIST Core Baseline in 8259, including sample text illustrating the general direction taken in 8259. A link to the full NISTIR 8259 draft is shown in Annex D: Informative References.

| CATEGORY | SUB-CATEGORY | MAPS TO |
|---------------------------------------|---|--|
| Secure Device Capabilities - Baseline | Device Identifiers | [NISTIR 8259] Table 1 row 1, Device Identification: The IoT device can be uniquely identified logically and physically. |
| Secure Device Capabilities - Baseline | Secured Access | [NISTIR 8259] Table 1 row 2, Device Configuration: The IoT device's software and firmware configuration 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. |
| Secure Device Capabilities - Baseline | Data In Transit Is Protected | [NISTIR 8259] 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 8259] 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 | |
| Secure Device Capabilities - Baseline | Event Logging | [NISTIR 8259] Table 1 row 6, Cybersecurity Event Logging: The IoT device can log cybersecurity events and make the logs accessible to authorized entities only. |
| Secure Device Capabilities - Baseline | Cryptography | [NISTIR 8259] 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 8259] Table 1 row 5, Software and Firmware Update: The IoT device's software and firmware can be updated by authorized entities only using a secure and configurable mechanism. |
| Secure Device Capabilities - Baseline | Reprovisioning | [NISTIR 8259] Table 1 row 2, Device Configuration: The IoT device's software and firmware configuration 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. |

| CATEGORY | SUB-CATEGORY | MAPS TO |
|---|---|--|
| Product Lifecycle Management | Vulnerability Submission and Handling Process | <p>[NISTIR 8259] Section 7: Manufacturers accepting and responding to vulnerability reports helps customers maintain the cybersecurity of their IoT devices as new threats emerge. SSDF practices:</p> <ul style="list-style-type: none"> • RV.1, Identify and Confirm Vulnerabilities on an Ongoing Basis • RV.2, Assess and Prioritize the Remediation of All Vulnerabilities • RV.3, Analyze Vulnerabilities to Identify Their Root Causes |
| Product Lifecycle Management | EoL/EoS Updates and Disclosure | [NISTIR 8259] Section 6: Support and Lifespan Expectations |
| Produce Lifecycle Management | Device Intent Documentation | [NISTIR 8259] Section 6: Sufficient information on the IoT device's operational characteristics so they can adequately secure the device (e.g., make information on characteristics available on a website...). |
| Secure Capabilities - Phase In Over Time | Device Intent Signaling | [NISTIR 8259] Section 6: Sufficient information on the IoT device's operational characteristics so they can adequately secure the device (e.g., ...use a standard protocol so devices can provide basic information to authorized parties). |
| Secure Capabilities - Phase In Over Time | Device Network Onboarding | |
| Additional IoT Device Security Capabilities and Practices | Secure Development Lifecycle | Section 7, Secure Development Practices for IoT Devices |
| Additional IoT Device Security Capabilities and Practices | Hardware Rooted Security | Section 5.1, Device Specifications: Use hardware-based cybersecurity features. An example is having a hardware root of trust that provides trusted storage for cryptographic keys and enables performing secure boots and confirming device authenticity. |
| 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 | [NISTIR 8259] Section 6: Cybersecurity Information to Provide to Customers |
| Additional IoT Device Security Capabilities and Practices | Least Functionality | [NISTIR 8259] Section 3.2, Device Cybersecurity Features: Do not include unneeded features provided by hardware, firmware, and/or the operating system; if the inclusion of such features cannot be avoided, ensure they can be disabled to prevent misuse and exploitation. |

| CATEGORY | SUB-CATEGORY | MAPS TO |
|---|-------------------------|--|
| Additional IoT Device Security Capabilities and Practices | Physical Access Control | [NISTIR 8259] Section 3.2, Device Cybersecurity Features: ...if a device has local interfaces on its external housing and the device is likely to be deployed in public areas, possible approaches include offering a tamper-resistant enclosure to prevent physical access to the interfaces, and offering a configuration option that logically disables the interfaces. |
| Additional IoT Device Security Capabilities and Practices | Best Current Practices | |

19 | Annex O: 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 | SUB-CATEGORY | MAPS TO |
|---------------------------------------|---|---|
| Secure Device Capabilities - Baseline | Device Identifiers | |
| Secure Device Capabilities - Baseline | Secured Access | [DCMS] 1. No default passwords All IoT device passwords shall be unique and not resettable to any universal factory default value |
| Secure Device Capabilities - Baseline | Data In Transit Is Protected | [DCMS] 4. Securely store credentials and security-sensitive data Any credentials shall be stored securely within services and on devices. Hard-coded credentials in device software are not acceptable. [DCMS] 5. Communicate securely Security-sensitive data, including any remote management and control, should be encrypted in transit, appropriate to the properties of the technology and usage. All keys should be managed 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 Data input via user interfaces and transferred via application programming interfaces (APIs) or between networks in services and devices shall be validated. |
| Secure Device Capabilities - Baseline | Event Logging | |
| Secure Device Capabilities - Baseline | Cryptography | |
| Secure Device Capabilities - Baseline | Patchability | [DCMS] 3. Keep software updated Software components in internet-connected devices should be securely updateable. Updates shall be timely and should not impact on the functioning of the device. An end-of-life policy shall be published for end-point devices which explicitly states the minimum length of time for which a device will receive software updates and the reasons for the length of the support period. The need for each update should be made clear to consumers and an update should be easy to implement. For constrained devices that cannot physically be updated, the product should be isolatable and replaceable. |

| CATEGORY | SUB-CATEGORY | MAPS TO |
|---|---|---|
| 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 All companies that provide internet-connected devices and services shall provide a public point of contact as part of a vulnerability disclosure policy in order that security researchers and others are able to report issues. Disclosed vulnerabilities should be acted on in a timely manner. |
| 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 | [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 | 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 Resilience should be built in to IoT devices and services where required by their usage or by other relying systems, taking into account the possibility of outages of data networks and power. As far as reasonably possible, IoT services should remain operating and locally functional in the case of a loss of network and should recover cleanly in the case of restoration of a loss of power. Devices should be able to return to a network in a sensible state and in an orderly fashion, rather than in a massive scale reconnect. |
| Additional IoT Device Security Capabilities and Practices | Secure Toolchains | |
| Additional IoT Device Security Capabilities and Practices | Software Transparency and Bill of Materials | |



| CATEGORY | SUB-CATEGORY | MAPS TO |
|---|-------------------------|--|
| Additional IoT Device Security Capabilities and Practices | Least Functionality | [DCMS] 6. Minimise exposed attack surfaces All devices and services should operate on the ‘principle of least privilege’; unused ports should be closed, hardware should not unnecessarily expose access, services should not be available if they are not used and code should be minimised to the functionality necessary for the service to operate. Software should run with appropriate privileges, taking account of both security and functionality. |
| Additional IoT Device Security Capabilities and Practices | Physical Access Control | |
| Additional IoT Device Security Capabilities and Practices | Best Practices | |

20 | Annex P: Mapping to UL MCV 1376 — Security Capabilities Verified

| CATEGORY | SUB-CATEGORY | MAPS TO |
|---------------------------------------|--------------------|--|
| Secure Device Capabilities - Baseline | Device Identifiers | None. |
| Secure Device Capabilities - Baseline | Secured Access | <p>[UL] 2.1 System defaults such as password, certificates, and/or cryptographic keys must be changed on initial setup</p> <p>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.</p> <p>[UL] 2.2 Password Policy</p> <p>Passwords 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 passwords are supposed to protect. Many attacks on devices are based on exploiting insecure, or default, password values.</p> <p>Minimum levels for password security to sensitive services must be enforced, such that there is less than a 1 in 100,000 chance that any guess will be correct and that attempts to brute force the password domain in the device cannot be performed in less than 24 hours. These protections may include combinations of password strength and ‘back off’ timers on any password entry mechanisms to slow entry during high volume password entry attempts. System designers should consider the needs of customers to re-enter incorrect passwords cause through typographic errors, along with the need (or lack thereof) to support many hundreds or thousands of password entry attempts within a relatively short (e.g., 24 hour) period.</p> <p>[UL] 2.3 Sensitive data must be protected against exposure and unauthenticated modification</p> <p>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.</p> <p>[UL] 3.1 Communication and debug ports must be protected against misuse</p> <p>Often devices will come with some interfaces that are either specifically designed, or can be used, for ‘debugging’ purposes. Additionally, all devices must of course have methods for communication. Such ports may be external or internal, allow for remote or local-only access, but all must be secured to prevent exploit. 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. Alternatively, a device may have remote communications — such as Wi-Fi, Ethernet, or others — which allows for data to be routed into and out of the device.</p> <p>Access to such ports therefore needs not be physical, but they may contain vulnerabilities or weaknesses that can be exploited to bypass protections in the device, expose customer PII, or install malware.</p> <p><i>(continued next page)</i></p> |

| CATEGORY | SUB-CATEGORY | MAPS TO |
|---------------------------------------|------------------------------|---|
| Secure Device Capabilities - Baseline | Secured Access | <p>The vendor must maintain a comprehensive list of all interfaces that the device supports — both physical and logical/protocol. This list must outline what access is provided across each of these interfaces, and how misuse of these interfaces and features is prevented through the design and implementation of the system.</p> <p>[UL] 4.1 Sensitive services must require authentication and ensure the confidentiality and integrity of data</p> <p>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 confidentiality of any customer data during transport.</p> |
| Secure Device Capabilities - Baseline | Data In Transit Is Protected | <p>[UL] [UL] 2.4 Industry standard cryptographic algorithms must be used for security services</p> <p>Cryptography has advanced to a point where there are common, standardized algorithms which are known to provide strong protection of data when correctly implemented. 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.</p> |
| Secure Device Capabilities - Baseline | Data At Rest Is Protected | <p>[UL] 2.3 Sensitive data must be protected against exposure and unauthenticated modification</p> <p>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.</p> <p>Additionally, 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 (eg 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.</p> <p>It is best practice to use a single ‘housekeeping’ key that is used as a master key for storage and protection of other sensitive data. Of course, this single key must then be stored securely, but it ensures that other data encrypted with this key can be maintained with lesser security, and may be transmitted across external busses without risk. The datagram format for the encryption of this data should accommodate for the type of data being encrypted — for example so that a simple password encrypted under the housekeeping key cannot be substituted for a more complex cryptographic key.</p> <p>Housekeeping keys must be stored securely, and never exposed in external memory or busses. It is a later requirement that the software which has the privilege to access this housekeeping key is executed at a more secure privilege to any code that has access to external ports or interfaces. Such keys must also be unique per device, as per the requirements of A.2.1.</p> |

| CATEGORY | SUB-CATEGORY | MAPS TO |
|---------------------------------------|---|--|
| Secure Device Capabilities - Baseline | Industry Accepted Protocols are Used for Communications | <p>[UL] 6.5 Switched or wireless connections must allow for the use of an industry standard security protocol (such as TLS)</p> <p>A formal security protocol is essential when communicating over remote or wireless connections. It is a requirement that systems allow for the use of an industry standard ‘best practice’ public protocol (such as TLS). A proprietary protocol that has been designed with the light weight needs of the IoT space in mind may also be implemented, but customers must be provided with the ability to choose which protocol they wish to use, and any proprietary protocols implemented may require the device to undergo more detailed testing. It should be noted that this requirement does not mandate that all communications must be performed using this protocol — for example, a light bulb may allow for the changing of brightness without implementing encryption of the command — but the protocol must be available for use, and must always be used for any security sensitive communications.</p> |
| Secure Device Capabilities - Baseline | Data Validation | <p>[UL] 4.4 No direct execution of scripts/commands</p> <p>Functionality that allows for the direct execution of scripts or commands by the device or system can often be exploited by a malicious party. Such functionality should not be natively supported, and any methods for the customer to supply executable code or scripts must be parsed and sanitized to ensure that it does not expose weaknesses or allow for the exploitation of security vulnerabilities in the system.</p> |
| Secure Device Capabilities - Baseline | Event Logging | <p>[UL] 3.8 Logging and error messages must not expose sensitive data without authentication</p> <p>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.</p> <p>It is also strongly recommended that any sensitive data that is logged is secured with cryptography (eg through encryption using a public key on the device). Any upload or exfiltration of customer 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.</p> <p>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 (eg 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 BCrypt function).</p> |
| Secure Device Capabilities - Baseline | Cryptography | <p>[UL] 2.4 Industry standard cryptographic algorithms must be used for security services</p> <p>Cryptography has advanced to a point where there are common, standardized algorithms which are known to provide strong protection of data when correctly implemented. 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.</p> |

| CATEGORY | SUB-CATEGORY | MAPS TO |
|---------------------------------------|----------------|--|
| Secure Device Capabilities - Baseline | Patchability | <p>[UL] 1.1 Software updates must be supported, using network or wireless interfaces where available</p> <p>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.</p> <p>[UL] 1.3 Software updates must be cryptographically authenticated, and provide anti-roll back features</p> <p>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 prevent its normal operation.</p> <p>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 firmware image, which can be checked by the original firmware (or bootloader of the device) prior to installation. Using a digital signature based on a public key algorithm (such as RSA, or DSA) ensures that the devices themselves don’t need the part of the key (the private or secret key) that is used to generate the authentication data.</p> <p>Where a symmetric key system — such as a (H)MAC — is used, the secret key in each device must be unique per device. Otherwise once the firmware of one device is exposed (eg through a physical attack on that one device), a valid firmware 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.</p> <p>It is additionally required that the update implements ‘anti-rollback’ features — such as a ‘monotonic’ version number which is included in each release (that is a version number that only increases with each version), which is also checked to ensure that any bad actor can’t just install a previous version of firmware; to ‘reinstate’ any otherwise patched vulnerabilities.</p> |
| Secure Device Capabilities - Baseline | Reprovisioning | <p>[UL] 4.2 Permanent erasure of sensitive data must be supported</p> <p>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.</p> |

| CATEGORY | SUB-CATEGORY | MAPS TO |
|--|---|--|
| Product Lifecycle Management | Vulnerability Submission and Handling Process | <p>[UL] 7.2 A vulnerability management and disclosure program must be maintained</p> <p>It has been noted above that it is impossible to find all bugs and vulnerabilities in software, and therefore 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:</p> <ul style="list-style-type: none"> • 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 the potential vulnerability, and any mitigating steps they can take whilst a patch is being created <p>[UL] 7.1 A documented process for the distribution of patches/updates must be maintained</p> <p>The final step to fixing a known vulnerability is to issue the patch to the customer/device. This must follow a clear process — which need not be complex, but must clearly outline the steps involved in approving, signing, and distributing the new code.</p> <p>This is required because it is often when there is a ‘rush’ to fix a problem that important security steps are missed, resulting in an even worse situation and more potential exposure of the systems which were being patched.</p> |
| Product Lifecycle Management | EoL/EoS Updates and Disclosure | <p>[UL] 7.2 A vulnerability management and disclosure program must be maintained</p> <p>It has been noted above that it is impossible to find all bugs and vulnerabilities in software, and therefore 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:</p> <ul style="list-style-type: none"> • 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 the potential vulnerability, and any mitigating steps they can take whilst a patch is being created <p>[UL] 7.1 A documented process for the distribution of patches/updates must be maintained</p> <p>The final step to fixing a known vulnerability is to issue the patch to the customer/device. This must follow a clear process — which need not be complex, but must clearly outline the steps involved in approving, signing, and distributing the new code.</p> <p>This is required because it is often when there is a ‘rush’ to fix a problem that important security steps are missed, resulting in an even worse situation and more potential exposure of the systems which were being patched</p> |
| Produce Lifecycle Management | Device Intent Documentation | |
| Secure Capabilities - Phase In Over Time | Device Intent Signaling | |

| CATEGORY | SUB-CATEGORY | MAPS TO |
|---|------------------------------|--|
| Secure Capabilities-Phase In Over Time | Device Network Onboarding | <p>[UL] 6.3 Connections to remote services must implement cryptographic authentication</p> <p>Remote access connections are especially vulnerable to attack and misuse, and so require special attention when it comes to security. Many interfaces are expected to use TLS for security, but TLS in and of itself is often not sufficient — so it is necessary not only to ensure the correct configuration of those protocols, but also that the authentication channel is ensuring that the customer can authenticate the server. This often requires validation of the complete TLS certificate, including organization, name and other fields, so that the interface cannot be intercepted and manipulated by a bad-actor who has their own TLS certificate. Other protocols may be similarly vulnerable, and would require other controls</p> |
| Additional IoT Device Security Capabilities and Practices | Secure Development Lifecycle | <p>[UL] 3.5 System software should be free of publically disclosed vulnerabilities</p> <p>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.</p> <p>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.</p> <p>[UL] 3.7 System software must be tested to check for undisclosed vulnerabilities</p> <p>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.</p> |
| Additional IoT Device Security Capabilities and Practices | Hardware Rooted Security | <p>[UL] 1.5 Device implements a hardware based root of trust for updates and boot authentication</p> <p>Although authenticating software updates is one important aspect of security, ensuring that any code is authenticated upon each boot of the device is also important. This ensures that even if changes are made to the executing code through some exploit, the changes cannot be made permanent and a reboot can be ensured to remove the malicious code.</p> |
| Additional IoT Device Security Capabilities and Practices | Time Distribution | |
| Additional IoT Device Security Capabilities and Practices | System Resiliency | <p>[UL] 4.3 Manual backup/override must be provided for safety related services</p> <p>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).</p> |

| CATEGORY | SUB-CATEGORY | MAPS TO |
|---|---|---|
| Additional IoT Device Security Capabilities and Practices | Secure Toolchains | <p>[UL] 3.4 Memory and compiler protections must be implemented</p> <p>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.</p> |
| Additional IoT Device Security Capabilities and Practices | Software Transparency and Bill of Materials | <p>[UL] 7.4 A ‘Software Composition List’ must be maintained</p> <p>It is an unfortunate truism that all software 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.</p> <p>To achieve this, it is necessary to create and keep up to date 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 out correctly. In this instance, if not all software is listed, you will not be able to securely maintain your device.</p> <p>Using this composition list, in concert with the vulnerability management program required below, 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 where necessary patched.</p> |
| Additional IoT Device Security Capabilities and Practices | Least Functionality | <p>[UL] 3.9 Systems must implement ‘least privilege’, or utilize hardware based features to protect sensitive code and data</p> <p>All software has vulnerabilities, and it is essential that ‘defense in depth’ measures are used to protect against the successful exploit of any newly discovered flaws. This leads to the implementation of ‘least privilege’ in systems, where software is assigned only the execution privilege and access rights that are sufficient and absolutely 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.</p> <p><i>(continued next page)</i></p> |



| CATEGORY | SUB-CATEGORY | MAPS TO |
|---|-------------------------|---|
| Additional IoT Device Security Capabilities and Practices | Least Functionality | <p>[UL] 3.6 Unwanted/unnecessary features are removed</p> <p>Often during the development of a product, features which were initially considered are removed from the scope, or existing code sets (such as 3rd party libraries or open source code) are used to speed up development. However, the more code that exists in a product, the more chances there are for that code to have bugs which can be exploited. Therefore, it is good security practice to remove unwanted or unnecessary features from code prior to using it in production devices, where these features have been deprecated during the development or where they are provide by default by external code (but can be removed or disabled if unwanted). This should be done at the source code level, to ensure that there is the smallest ‘attack surface’ possible in the shipped code files.</p> <p>[UL] 7.5 All protocols present in the device must be documented and justified</p> <p>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).</p> <p>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 on these interfaces. However, this requirement is designed to cover only physical and output-originating protocols — listening services that actively wait for connections over switched and wireless interfaces are covered under a separate requirement below.</p> <p>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 and so this exercise is an important one.</p> <p>[UL] 7.6 All services present in the device must be documented and justified</p> <p>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’.</p> <p>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.</p> <p>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.</p> |
| Additional IoT Device Security Capabilities and Practices | Physical Access Control | |

| CATEGORY | SUB-CATEGORY | MAPS TO |
|---|----------------|--|
| Additional IoT Device Security Capabilities and Practices | Best Practices | <p>[UL] 6.5 Switched or wireless connections must allow for the use of an industry standard security protocol (such as TLS)</p> <p>A formal security protocol is essential when communicating over remote or wireless connections. It is a requirement that systems allow for the use of an industry standard ‘best practice’ public protocol (such as TLS). , A proprietary protocol that has been designed with the light weight needs of the IoT space in mind may also be implemented, but customers must be provided with the ability to choose which protocol they wish to use, and any proprietary protocols implemented may require the device to undergo more detailed testing. It should be noted that this requirement does not mandate that all communications must be performed using this protocol — for example, a light bulb may allow for the changing of brightness without implementing encryption of the command — but the protocol must be available for use, and must always be used for any security sensitive communications.</p> <p>[UL] 6.4 Security protocols must implement secure defaults, and prevent downgrade attacks</p> <p>Many security protocols, such as TLS, allow for the use of insecure protocols and methods. Even when secure options are used, sometimes the connection can be forced to downgrade to a less secure option if it is not correctly configured.</p> <p>[UL] 6.2 Device must support industry accepted wireless security defaults for any Wi-Fi connections</p> <p>Where devices implement Wi-Fi connections, it is important that these devices do not force a reduction in the security of the customers Wi-Fi implementation. For example, a poorly designed system may force the customer to change from WPA2 security to using WEP, which is considered insecure.</p> |



21 | Sponsoring Organizations

About CSDE ▶ The Council to Secure the Digital Economy (CSDE) brings together companies from across the information and communications technology (ICT) sector to combat increasingly sophisticated and emerging cyber threats through collaborative actions. Members include Akamai, AT&T, CA Technologies, CenturyLink, Cisco, Ericsson, IBM, Intel, NTT, Oracle, Samsung, SAP, Telefonica and Verizon. CSDE is coordinated by USTelecom and the Consumer Technology Association.

About USTelecom ▶ USTelecom is the premier trade association representing service providers and suppliers for the telecom industry. Its diverse member base ranges from large publicly traded communications corporations to small companies and cooperatives— all providing advanced communications service to both urban and rural markets.

About the Consumer Technology Association ▶ The Consumer Technology Association (CTA)[™] is the trade association representing the \$377 billion U.S. consumer technology industry, which supports more than 15 million U.S. jobs. More than 2,200 companies — 80 percent are small businesses and startups; others are among the world's best-known brands — enjoy the benefits of CTA membership including policy advocacy, market research, technical education, industry promotion, standards development and the fostering of business and strategic relationships. CTA also owns and produces CES[®] — the world's gathering place for all who thrive on the business of consumer technologies. Profits from CES are reinvested into CTA's industry services.

22 | Endnotes

- 1 Daniel Newman, *The Top 8 IoT Trends for 2018*, Forbes (Dec. 19, 2017), <https://www.forbes.com/sites/danielnewman/2017/12/19/the-top-8-iot-trends-for-2018/#2523096867f7> (citing HIS Markit IoT Trend Watch 2018, available at <https://ihsmarkit.com/industry/telecommunications.html>); See also Gartner, *Gartner Says 8.4 Billion Connected “Things” Will Be in Use in 2017, Up 31 Percent From 2016* (Feb. 7, 2017), <https://www.gartner.com/en/newsroom/press-releases/2017-02-07-gartner-says-8-billion-connected-things-will-be-in-use-in-2017-up-31-percent-from-2016>.
- 2 See, e.g., Catalin Cimpanu, *Sly Malware Author Hides Cryptomining Botnet Behind Ever-shifting Proxy Service*, ZDNet (Sept. 13, 2018), <https://www.zdnet.com/article/sly-malware-author-hides-cryptomining-botnet-behind-ever-shifting-proxy-service/> (“[B]otnets focused on cryptocurrency mining operations have been one of the most active forms of malware infections in 2018.”)
- 3 Sam Thielman and Chris Johnston, *Major Cyber Attack Disrupts Internet Service Across Europe and US*, The Guardian, (Oct. 21, 2016), <https://www.theguardian.com/technology/2016/oct/21/ddos-attack-dyn-internet-denial-service>.
- 4 Michael Newberg, *As Many as 48 Million Twitter Accounts Aren’t People, Says Study*, CNBC (Mar. 10, 2017), <https://www.cnbc.com/2017/03/10/nearly-48-million-twitter-accounts-could-be-bots-says-study.html>.
- 5 JP Buntinx, *Top 4 Largest Botnets to Date*, Null TX (Jan. 7, 2017).
- 6 <https://www.csis.org/analysis/extending-federal-cybersecurity-endpoint>
- 7 <https://www.catonetworks.com/blog/iot-security-standards-and-initiatives/>
- 8 European Commission, *EU negotiators agree on strengthening Europe’s cybersecurity*, Dec. 2018, http://europa.eu/rapid/press-release_IP-18-6759_en.htm
- 9 Japan Ministry of Economy, Trade and Industry, *METI Compiles Results of the Call for Public Comments on the Draft Cyber/Physical Security Framework*, https://www.meti.go.jp/english/press/2018/1001_002.html
- 10 UK Department of Digital, Culture, Media & Sport, *Code of Practice for Consumer IoT Security*, Oct. 2018, https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/773867/Code_of_Practice_for_Consumer_IoT_Security_October_2018.pdf
- 11 UK Department of Digital, Culture, Media & Sport, *Consultation on regulatory proposals on consumer IoT security*, <https://www.gov.uk/government/consultations/consultation-on-regulatory-proposals-on-consumer-iot-security>, May 2019
- 12 NIST, *NIST Cybersecurity for IoT Program*, retrieved Mar. 2019, <https://www.nist.gov/programs-projects/nist-cybersecurity-iot-program>
- 13 E.g., <https://www.publicknowledge.org/press-release/new-public-knowledge-paper-proposes-security-shield-label-to-support-sustainable-cybersecurity>.
- 14 See the informative references section for industry voluntary consensus standards and best practices.
- 15 Available at <https://nvlpubs.nist.gov/nistpubs/ir/2019/NIST.IR.8228.pdf>. Further work is available at https://www.nist.gov/sites/default/files/documents/2019/02/01/final_core_iiot_cybersecurity_capabilities_baseline_considerations.pdf.
- 16 Available at <https://nvlpubs.nist.gov/nistpubs/ir/2019/NIST.IR.8259-draft.pdf>.
- 17 See Rob van der Meulen, *Gartner Says 8.4 Billion “Things” Will Be in Use in 2017, Up 31 Percent From 2016*, Gartner (Feb. 7, 2017), <https://www.gartner.com/en/newsroom/press-releases/2017-02-07-gartner-says-8-billion-connected-things-will-be-in-use-in-2017-up-31-percent-from-2016> (reporting that the majority of the IoT market is comprised of consumer applications, which are generally less complex devices).
- 18 Or software-hardware hybrid.
- 19 In this case, a user may refer to the consumer using a device, a technician responsible for installation or maintenance, an authorized employee in a managed environment, etc.
- 20 Transport Layer Security version 1.3, see <https://datatracker.ietf.org/doc/rfc8446/>.
- 21 *Prohibiting Secure Sockets Layer (SSL) Version 2.0*, see <https://tools.ietf.org/html/rfc6176>.
- 22 See, e.g., [https://www.owasp.org/index.php/Cross-site_Scripting_\(XSS\)](https://www.owasp.org/index.php/Cross-site_Scripting_(XSS))
- 23 Updateability may also be referred to as agility.
- 24 For more information on coordinated vulnerability disclosure, see *The CERT Guide to Coordinated Vulnerability Disclosure*, <https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=503330>. ISO 30111 (internal processes) and ISO 29147 (disclosure) are good references as well on this topic, as well as Center for Cybersecurity Policy and Law’s *Improving Hardware Component Vulnerability Disclosure*, <https://centerforcybersecuritypolicy.org/improving-hardware-component-vulnerability-disclosure>.
- 25 See <https://datatracker.ietf.org/doc/rfc8520/>
- 26 OMA Device Management, see http://www.openmobilealliance.org/wp/overviews/dm_overview.html
- 27 See <https://www.broadband-forum.org/download/TR-069.pdf>
- 28 IoTSense: Behavioral Fingerprinting of IoT Devices, Colorado State University, April 2018, <https://arxiv.org/pdf/1804.03852.pdf>
- 29 For example, <https://www.nccoe.nist.gov/sites/default/files/library/project-descriptions/iiot-ddos-project-description-final.pdf>
- 30 See also <https://github.com/w3c/wot-security-testing-plan> for an open-source IoT security testing plan, including functional and adversarial testing, and a security test plan framework.
- 31 See also <https://www.ul.com/resources/ul-cybersecurity-assurance-program-ul-cap> for the UL Cybersecurity Assurance Program.
- 32 NIST, essay referenced in blog post Let’s talk about IoT device security, Feb. 2019, https://www.nist.gov/sites/default/files/documents/2019/02/01/final_core_iiot_cybersecurity_capabilities_baseline_considerations.pdf
- 33 Ibid.
- 34 See *EU Agency for Cybersecurity Baseline Security Recommendations for IoT*, <https://www.enisa.europa.eu/publications/baseline-security-recommendations-for-iiot>

