MCMC MTSFB TC G044:2023

TECHNICAL CODE

INTERNET OF THINGS (IOT) - BASELINE SECURITY REQUIREMENTS FOR CONSUMER DEVICES

Developed by

Malaysian Technical Standards Forum Bhd

Registered by

MCMC

Registered date: 31 October 2023

MCMC MTSFB TC G044:2023

Development of technical codes

The Communications and Multimedia Act 1998 (Laws of Malaysia Act 588) ('the Act') provides for a Technical Standards Forum designated under section 184 of the Act or the Malaysian Communications and Multimedia Commission ('the Commission') to prepare a technical code. The technical code prepared pursuant to section 185 of the Act shall consist of, at least, the requirements for network interoperability and the promotion of safety of network facilities.

Section 96 of the Act also provides for the Commission to determine a technical code in accordance with section 55 of the Act if the technical code is not developed under an applicable provision of the Act and it is unlikely to be developed by the Technical Standards Forum within a reasonable time.

In exercise of the power conferred by section 184 of the Act, the Commission has designated the Malaysian Technical Standards Forum Bhd ('MTSFB') as a Technical Standards Forum which is obligated, among others, to prepare the technical code under section 185 of the Act.

A technical code prepared in accordance with section 185 shall not be effective until it is registered by the Commission pursuant to section 95 of the Act.

For further information on the technical code, please contact:

Malaysian Communications and Multimedia Commission (MCMC)
MCMC Tower 1
Jalan Impact
Cyber 6
63000 Cyberjaya
Selangor Darul Ehsan
MALAYSIA

Tel: +60 3 8688 8000
Fax: +60 3 8688 1000
http://www.mcmc.gov.my

OR

Malaysian Technical Standards Forum Bhd (MTSFB)
MCMC Tower 2
Jalan Impact
Cyber 6
63000 Cyberjaya
Selangor Darul Ehsan
MALAYSIA

Tel: +60 3 8680 9950
Fax: +60 3 8680 9940
http://www.mtsfb.org.my

Contents

Committee representation

This technical code was developed by the Internet of Things and Smart Sustainable Cities Working Group of the Malaysian Technical Standards Forum Bhd (MTSFB), which consists of representatives from the following organisations:

Cyberview Sdn Bhd

Favoriot Sdn Bhd

Maxis Broadband Sdn Bhd

Sunway University College Sdn Bhd

Telekom Malaysia Berhad

UCSI Education Sdn Bhd

Universiti Malaya

Universiti Putra Malaysia

Universiti Sains Islam Malaysia

Universiti Teknologi MARA

Foreword

This technical code for Internet of Things (IoT) - Baseline Security Requirements for Consumer Devices ('Technical Code') was developed pursuant to Section 185 of the Communications and Multimedia Act 1998 (Laws of Malaysia Act 588) by the Internet of Things and Smart Sustainable Cities Working Group of the Malaysian Technical Standards Forum Bhd (MTSFB).

This Technical Code shall continue to be valid and effective from the date of its registration until it is replaced or revoked.

INTERNET OF THINGS (IOT) - BASELINE SECURITY REQUIREMENTS FOR CONSUMER DEVICES

0. Introduction

The Internet of Things has been touted as a technological advancement that has profoundly affected human life. Consumer Internet of Things (IoT) refers to the incorporation of IoT into typical consumer applications and devices. Consumer IoT applications can range from the relatively simple and cheap, such as personal fitness devices, to the sophisticated and expensive, such as smart home automation technology. Hence, consumer IoT use cases, devices, and applications are highly diversified.

One of the use cases for Consumer IoT is the smart home. Smart home devices consist of those IoT devices or gadgets that can assist users with their daily activities at home. For example, smart bulbs would allow homeowners to switch on and off the lights even while they are not at home, and smart locks will replace the need for physical keys, allowing homeowners to enter their houses just by using their fingerprints or passcodes. However, there is growing concern that these novel technologies may pose risks to safety.

In this regard, there is an urgent need for the development of Baseline Security Requirements for Consumer IoT which can act as the foundation and guidance for the security requirements in consumer IoT devices.

1. Scope

The Technical Code specifies baseline security and data protection requirements for consumer IoT devices that are connected to network infrastructure such as the Internet or home network and their interactions with associated services. This Technical Code provides basic guidance, through examples and explanatory text, for organisations involved in the development and manufacturing of consumer IoT devices on how to implement these requirements.

Non-consumer IoT devices used in sectors such as manufacturing and healthcare are not in the scope of this document.

2. Normative references

The following normative references are indispensable for the application of this Technical Code. For dated references, only the edition cited applies. For undated references, the latest edition of the normative references (including any amendments) applies.

MCMC MTFSB TC G042, *Information and Network Security- Malaysia Critical Security Controls (MYCSC)*

ISO/IEC 29147, *Information technology, Security techniques, Vulnerability Disclosure*

PDP Code of Practice, *For Licensees Under the Communications and Multimedia Act 1998*

MyVAC-3-GUI-3-IOT-v1, *Guidelines for Secure Internet of Things (IoT)*

NIST Special Publication 800-63B

ETSI TR 103 331 CYBER; Structured Threat Information Sharing

3. Abbreviations

For the purposes of this Technical Code, the abbreviations in Annex A apply.

4. Terms and definitions

For the purposes of this Technical Code, the following definitions apply:

4.1 Administrator

User who has the highest privilege level possible for a user of the device, which can mean they are able to change any configuration related to the intended functionality.

4.2 Associated services

Digital services that, together with the device, are part of the overall consumer IoT product and that are typically required to provide the product's intended functionality.

Refer to example in Table B.1 Annex B.

4.3 Authentication mechanism

A method used to prove the authenticity of an entity where an entity can be either a user or machine.

Refer to example in Table B.1 Annex B.

4.4 Authentication value

The individual value of an attribute used by an authentication mechanism.

Refer to example in Table B.1 Annex B.

4.5 Best practice cryptography

Cryptography that is suitable for the corresponding use case and has no indications of a feasible attack with current readily available techniques.

NOTES:

1. This does not refer only to the cryptographic primitives used but also to implementation, key generation and handling of keys.

2. Multiple organisations, such as Standards Development Organisations (SDOs) and public authorities, maintain guides and catalogues of cryptographic methods that can be used.

Refer to example in Table B.1 Annex B.

4.6 Constrained device

A device that has physical limitations in either its ability to process data or its ability to communicate data or its ability to store data or its ability to interact with the user due to restrictions that arise from its intended use.

NOTES:

1. Physical limitations can be due to power supply, battery life, processing power, physical access, limited functionality, limited memory, or limited network bandwidth. These limitations can require a constrained device to be supported by another device, such as a base station or companion device.

2. A device that has a wired power supply and can support Internet Protocol (IP)-based protocols and the cryptographic primitives used by those protocols is not considered as a constrained device.

Refer to example in Table B.1 Annex B.

4.7 Consumer

Natural person who is acting for purposes that are outside his or her trade, business, craft or profession. A person or a group who intends to use the device for personal, social, family, household, and similar needs, which is not directly related to business activities.

NOTE: Organisations, including businesses of any size, use consumer IoT. For example, smart Televisions (TV) are frequently deployed in meeting rooms, and home security kits can protect the premises of small businesses.

4.8 Consumer IoT Devices

Network-connected and network-connectable devices that have relationships to associated services and are used by the consumer typically in the home or as electronic wearables such as smart TV, smart fridge or Google Home.

NOTES:

1. Consumer IoT devices are commonly also used in business contexts. These devices remain classified as consumer IoT devices.

2. Consumer IoT devices are often available for the consumer to purchase in retail environments. Consumer IoT devices can also be commissioned and/or installed professionally.

4.9 Critical security parameter

Security-related secret information whose disclosure or modification can compromise the security of a security module.

Refer to example in Table B.1 Annex B.

4.10 Debug interface

Physical interface used by the manufacturer to communicate with the device during development or to perform triage of issues with the device and that is not used as part of the consumer-facing functionality.

Refer to example in Table B.1 Annex B.

4.11 Defined support period

Length of time, expressed as a period or by an end-date, for which a manufacturer will provide security updates.

NOTE: This definition focuses on security aspects and no other aspects related to product support such as warranty.

4.12 Device manufacturer

Entity that creates an assembled final consumer IoT product, which is likely to contain the products and components of many other suppliers.

4.13 Factory default

State of the device after factory reset or after final production or assembly.

NOTE: This includes the physical device and software including firmware that is present in it after assembly.

4.14 Initialisation

Process that activates the network connectivity of the device for operation and optionally sets authentication features for a user or for network access.

4.15 Initialised state

State of the device after initialisation.

4.16 IoT product

Consumer IoT device and its associated services.

4.17 Isolable

Able to be removed from the network it is connected to, where any functionality loss caused is related only to that connectivity and not to its main function; alternatively, able to be placed in a self-contained environment with other devices only if the integrity of the devices within that environment can be ensured.

Refer to example in Table B.1 Annex B.

4.18 Logical interface

Software implementation that utilises a network interface to communicate over the network via channels or ports.

4.19 Manufacturer

Relevant economic operator in the supply chain including the device manufacturer and others (e.g. importers, distributors, integrators, component and platform providers, software providers, IT and telecommunications service providers, managed service providers and providers of associated services).

4.20 Network interface

Physical interface that can be used to access the functionality of consumer IoT via a network.

4.21 Owner

User who owns or who purchased the device.

4.22 Personal data

Any information relating to an identified or identifiable natural person.

4.23 Physical interface

Physical port or air interface such as radio, audio or optical that is used to communicate with the device at the physical layer.

Refer to example in Table B.1 Annex B.

4.24 Public security parameter

Security related public information whose modification can compromise the security of a security module.

Refer to example in Table B.1 Annex B.

4.25 Remotely accessible

Intended to be accessible from outside the local network.

4.26 Security module

Set of hardware, software, and/or firmware that implements security functions.

Refer to example in Table B.1 Annex B.

4.27 Security update

Software update that addresses a security vulnerability that is either discovered by or reported to the manufacturer.

NOTE: A software update can be purely a security update if the severity of the vulnerability requires a higher priority fix.

4.28 Sensitive security parameters

Critical security parameters and public security parameters.

4.29 Software service

Software component of a device that is used to support its functionality.

Refer to example in Table B.1 Annex B.

4.30 Telemetry

Data from a device that can provide information to help the manufacturer identify issues or provide information related to device usage.

Refer to example in Table B.1 Annex B.

4.31 Unique per device

Unique for each individual device of a given product class or type.

4.32 User

Natural person or organisation.

5. Overview

Home-deployed consumer IoT often comprises a variety of constrained and non-constrained devices connected directly to the Local Area Network (LAN) via IP connectivity, such as Ethernet or Wireless Fidelity (Wi-Fi), or indirectly via a gateway or hub. This indirect LAN and Personal Area Network (PAN) connection will often utilise non-IP connectivity and Bluetooth. Afterwards, a router will connect the LAN to the Wide Area Network (WAN) (i.e. the Internet). In certain instances, however, a home device can connect directly to the WAN via non-IP or IP connections such as Global System for Mobile Communications (GSM) or Long Range Wide Area Network (LoRaWAN).

Consumer IoT devices frequently connect to the internet or to local services or are linked to such services. Associated services are included by the manufacturer or shall be installed as part of the start-up process. When the user installs a service or accesses external content, these instances would not be considered associated services.

Figure 1 is a realistic illustration of a general home architecture for deploying consumer IoT devices. The ‘home’ border approximates the extent of this Technical Code’s declared scope, which includes communication with associated services.

The following use cases illustrate how this setup would be used:

a) Two external services are accessible via the smart TV. The first is the Device Telemetry Service (DTS) which is an associated service that collects information from the smart TV, such as crash logs and usage data, with the user's consent, so that the manufacturer can resolve software bugs and prioritize the development of new features. The second external service occurs after the initialisation where the smart TV connects to a video sharing service via an application downloaded by the user. This video sharing service permits entertainment viewing via a third-party application that can be installed on the TV's operating system. This streaming service is not an associated service.

b) The gateway enables access to several constrained devices, such as an IEEE 802.15.4 mesh network and a light sensor for monitoring and managing the home. It connects to a cloud access service that allows users to control their smart locks and view sensor data remotely. This is an associated service.

c) The smart fridge is equipped with a web browser, allowing users to access news from the fridge display. The news website would not be an associated service.

d) The user uses the weather sensor to determine the outdoor temperature. As it is geographically distant from the residence, it cannot connect to the LAN. Instead, it interacts directly with the WAN using GSM. The service to which the weather sensor connects is an associated service.

e) Digital Voice Assistant Device (DVAD) may act as a bridge between the Internet and the various consumer IoT devices. It uses a voice-driven virtual assistance to communicate and activate smart home devices. The service to which the DVAD connects is an associated service. However, the website the consumer visits is not an associated service.

f) Fitness tracker device that is connected to its application in the smart phone via Bluetooth Low Energy (BLE) to store and monitor the users’ data. The database obtained from the fitness tracker

application is stored in the cloud for monitoring and analysis. The fitness tracker application is not an associated service.

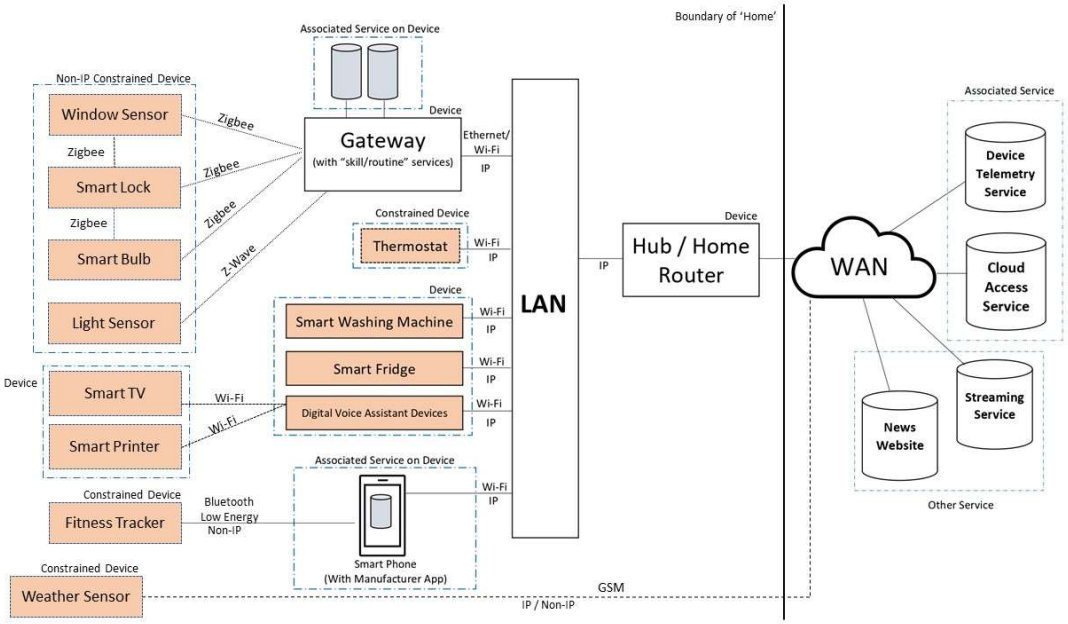

Figure 1: Overview of security baseline

Consumer IoT devices' hardware modules are another crucial security component. They are the 'brains' of the devices and may be found in every consumer IoT device. With many of these devices being deployed in public, it is important to include appropriate levels of security to prevent the devices from invasive, non-invasive or semi-invasive attacks. In order to protect the consumer IoT devices, it is crucial to ensure the security of its hardware modules. Embedded security also known as device security~~, and it~~ can help to protect the hardware modules against real threats such as Denial-of-Service (DoS) attacks, counterfeiting and device tampering.

The general block diagram of the internal hardware modules of consumer IoT devices is seen in Figure 2. It is made up of a few hardware submodules that work together to collect and analyse sensor data and activate the actuators or consumer devices. Common security vulnerability points are interfaces that are usually compromised by the attackers. The possibilities of being attacked can be reduced by minimising the exposed attack surfaces in the consumer IoT devices. ISO/IEC 29147 provides details of security vulnerabilities.

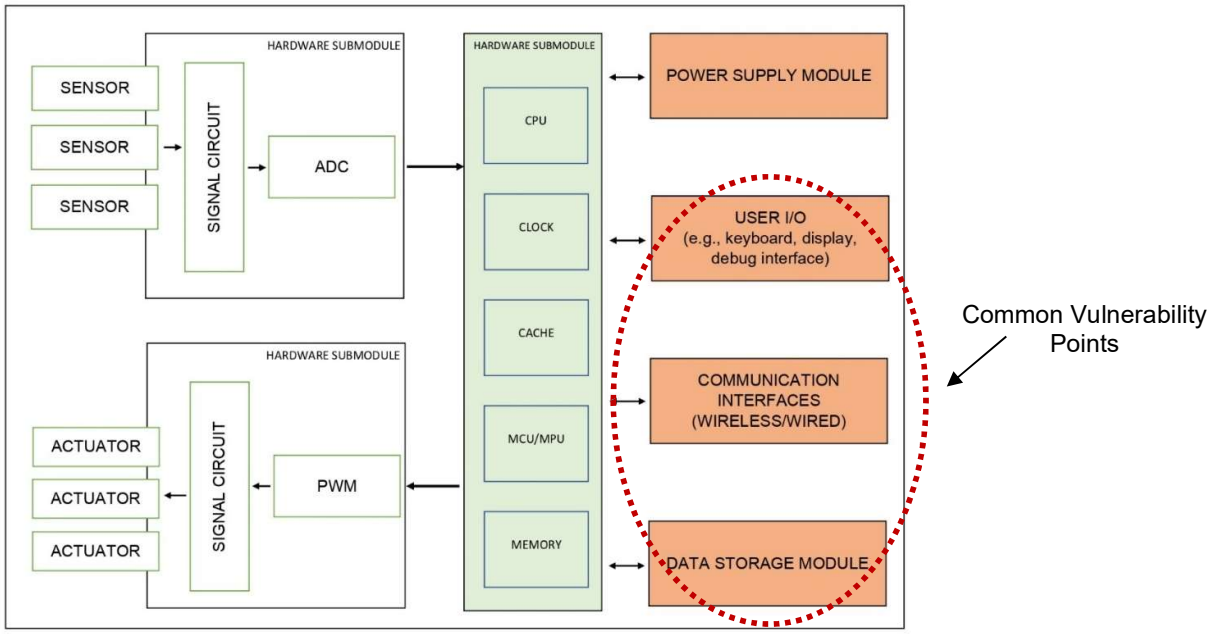

Figure 2: General block diagram of internal hardware modules of IoT consumer devices

6. Cyber security requirements for consumer IoT

6.1 No universal default passwords

No universal default passwords are an important requirement because default passwords that are easily guessable or derivable may weaken security.

a) Where passwords are used and, in any state, other than the factory default, all consumer IoT device passwords shall be unique per device or defined by the user.

Many consumer IoT devices are sold with universal default usernames and passwords for the user to get through to the network. Continued usage of universal default values has been the source of many security issues in IoT and the practice should be discontinued. This can be achieved by the use of pre-installed passwords that are unique per device and/or by requiring the user to choose a password that follows best practice as part of initialisation, or by some other method that does not use passwords.

Multi-factor authentication, such as a password together with a One-Time Password (OTP) can be used to better protect the device or an associated service. Device security can be further strengthened by having unique and immutable identities.

Refer to example in Table C.1 Annex C.

There are many mechanisms used for performing authentication, and passwords are not the only mechanism for authenticating a user to a device. However, if they are used, following best practice on passwords according to NIST Special Publication 800-63B is encouraged. Using passwords for machine-to-machine authentication is generally not appropriate.

b) Where pre-installed unique per device passwords are used, these shall be generated with a mechanism that reduces the risk of automated attacks against a class or type of device.

As a counterexample, passwords with incremental counters (e.g. ‘password1’, ‘password2’, etc) are easily guessable. Further, using a password that is related in an obvious way to public information (sent over the air or within a network), such as Media Access Control (MAC) address or Wi-Fi Service Set Identifier (SSID), can allow for password retrieval using automated means.

Refer to example in Table C.1 Annex C.

c) Authentication mechanisms used to authenticate users against a device should use best practice cryptography, appropriate to the properties of the technology, risk and usage.

d) Where a user can authenticate against a device, the device shall provide to the user or an administrator a simple mechanism to change the authentication value used.

An authentication mechanism used for authenticating users, whether a fingerprint, password or other token, needs to have its value changeable. This is easier when this mechanism is part of the normal usage flow of the device.

Refer to example in Table C.1 Annex C.

e) When the device is not a constrained device, it shall have a mechanism available which makes brute-force attacks on authentication mechanisms via network interfaces impracticable.

Attacks that perform ‘credential stuffing’ or exhaust an entire key-space should be detected by the consumer IoT device and defended against, whilst guarding against related threats of ‘resource exhaustion’ and Denial-of-Service (DoS) attacks.

Refer to example in Table C.1 Annex C.

6.2 Managing reports of vulnerabilities

6.2.1 Disclosure policy publicly available

The manufacturer shall make its vulnerability disclosure policy publicly available. The policy shall include disclosure of the following as a minimum:

a) contact information for the reporting of issues; and

b) information on timelines for:

i) initial acknowledgement of receipt of report; and

ii) status updates until the resolution of the reported issues.

The vulnerability disclosure policy should clearly specify the process through which security researchers and others are able to report issues. The policy should be updated as necessary to ensure transparency and clarity in the dealings of the manufacturer with security researchers, and vice versa.

In establishing its processes for security vulnerability disclosure, the manufacturer should give consideration to Coordinated Vulnerability Disclosure (CVD), which is a set of standardised processes

for dealing with disclosures about potential security vulnerabilities and to support the remediation of these vulnerabilities. Reference should be made to ISO/IEC 29147 which provides requirements and recommendations on vulnerability disclosure.

CVD provides manufacturers with a framework to manage the processes of vulnerability disclosure. It gives security researchers an avenue to inform manufacturers of security issues, puts manufacturers ahead of the threat of malicious exploitation and gives them an opportunity to respond to and resolve vulnerabilities in advance of a public disclosure.

6.2.2 Disclosed vulnerabilities rectification time

Disclosed vulnerabilities should be acted on in a timely manner. A timely manner for acting on vulnerabilities varies considerably and is incident-specific. As a general guide, the vulnerability process should be completed within 90 days for a software solution, including availability of patches and notification of the issue. A hardware fix can take considerably longer to address than a software fix. Additionally, a fix that has to be deployed to devices can take time to roll out compared with a server software fix.

6.2.3 Vulnerabilities monitoring

Manufacturers should continually monitor, identify and rectify security vulnerabilities within products and services they sell, produce, have produced and services they operate during the defined support period.

Manufacturers should create and maintain a list of all open source and third-party software components and their sub-components to be able to monitor for product vulnerabilities. Appropriate tools should be used to scan source code and binaries and build a so-called Software Bill of Materials (SBOM), which identifies third party components and the versions used in the product. This information should be used to monitor ~~for~~ the associated security and licensing risks of each identified software component.

Vulnerabilities should be reported directly to the affected stakeholders in the first instance. If that is not possible, vulnerabilities should be reported to the relevant authorities. Guidance on Coordinated Vulnerability Disclosure is available from the IoT Security Foundation which references ISO/IEC 29147.

This should be performed for devices within their defined support period. However, manufacturers can continue this outside that period and release security updates to rectify vulnerabilities.

Manufacturers that provide IoT products have a duty of care to consumers and third parties who can be harmed by their failure to have a CVD programme in place.

Disclosures can comprise different approaches depending on the circumstances:

a) Vulnerabilities related to single products or services

   The problem is expected to be reported directly to the affected stakeholder (e.g. device manufacturer, IoT service provider or mobile application developer). The source of these reports can be security researchers or industry peers.

b) Systemic vulnerabilities

A stakeholder, such as a device manufacturer, can discover a problem that is potentially systemic. Whilst fixing it in the device manufacturer's own product is crucial, there is significant benefit to industry and consumers from sharing this information. Similarly, security researchers can also seek to report such systemic vulnerabilities. For systemic vulnerabilities, a wider scale response should be coordinated by the relevant competent industry body.

Cyber security threat information sharing can support organisations in developing and producing secure products according to ETSI TR 103 331.

6.3 Keep software updated

Manufacturers should develop and deploy security updates in a timely manner to protect customers and the wider technical ecosystem. It is good practice that all software is kept updated and well maintained.

6.3.3 to 6.3.12 is dependent upon an update mechanism being implemented, as per 6.3.1 or 6.3.2.

6.3.1 Updating of software components

Software components in consumer IoT devices shall be securely updated. However, not all software on a device is updateable.

Refer to example in Table C.1 Annex C.

NOTE: Managing software updates generally relies on communication of version information for software components between the device and the manufacturer.

6.3.2 Secure installation of updates

When the device is not a constrained device, it shall have a mechanism for the secure installation of updates.

Secure updating and secure installation means that there are adequate measures to prevent an attacker from misusing the update mechanism.

Update mechanisms can range from the device downloading the update directly from a remote server, transmitted from a mobile application or transferred over a USB or other physical interface. If an attacker compromises this mechanism, it allows for a malicious version of the software to be installed on the device.

Refer to example in Table C.1 Annex C.

6.3.3 Simple update

An update should be simple for the user to apply. If an update is difficult to apply, there is an increased risk that a user will repeatedly defer updating the device, leaving it in a vulnerable state.
The degree of simplicity depends on the design and intended usage of the device. An update that is simple to apply will be automatically applied, initiated using an associated service such as a mobile application or via a web interface on the device.

6.3.4 Automatic mechanisms for software updates

Automatic mechanisms should be used for software updates. Detection mechanisms such as watchdogs and the use of dual-bank flash or recovery partitions can ensure that the device returns to either a known good version or the factory state.

As part of automatic updates, security updates can be provided for devices in a preventative manner, which can remove security vulnerabilities before they are exploited. Managing this can be complex, especially if there are parallel associated service updates, device updates and other service updates to deal with. Therefore, a clear management and deployment plan is beneficial to the manufacturer, as is transparency to consumers about the current state of update support.

In many cases, publishing software updates involves multiple dependencies on other organisations such as manufacturers that produce sub-components. However, this is not a reason to withhold updates. It can be useful for the manufacturer to consider the entire software supply chain in the development and deployment of security updates.

It is advisable not to bundle security updates with more complex software updates, such as feature updates. A feature update that introduces new functionality can trigger additional requirements and delay delivery of the update to devices.

Refer to example in Table C.1 Annex C.

6.3.5 Periodic security updates

The device shall check after initialisation and periodically, whether security updates are available. For some devices, it can be more appropriate for the associated service, rather than the device, to perform such checks.

Refer to example in Table C.1 Annex C.

6.3.6 Automatic updates

If the device supports automatic updates and/or update notifications, these should be enabled in the initialised state and configurable so that the user can enable, disable, or postpone installation of security updates and/or update notifications.

It is important from a consumer rights and ownership perspective that the user is in control of whether or not they receive updates. There are good reasons why a user may choose not to update, including security updates. In addition, if an update is deployed and subsequently found to cause issues, manufacturers can ask users to not upgrade their software in order that those devices are not affected.

6.3.7 Secure update mechanism

The device shall use best practice cryptography to facilitate secure update mechanisms.

6.3.8 Timeliness of security updates

Security updates shall be implemented in a timely manner. Timeliness in the context of security updates can vary, depending on:

a) the particular issue and fix;

b) the ability to reach a device; or

c) constrained device considerations.

It is important that a security update that fixes a critical vulnerability i.e. one with potentially adverse effects of a large scale, is handled with appropriate priority by the manufacturer. Due to the complex structure of modern software and the ubiquity of communication platforms, multiple stakeholders can be involved in a security update.

Refer to example in Table C.1 Annex C.

6.3.9 Authenticity and integrity verification

To confirm that an update is valid, the device should verify the authenticity and integrity of the software updates. This can be done on the device. However, constrained devices can have power limitations that make performing cryptographic operations costly. In such cases, verification can be performed by another device that is trusted to perform this verification. The verified update would then be sent over a secure channel to the device. Performing verification of updates at a hub and then on the device, can reduce the risk of compromise.

It is good practice for a device to act upon the detection of an invalid and potentially malicious update. Beyond rejecting the update, it should report the incident to an appropriate service and/or inform the user. In addition, mitigating controls should be put in place to prevent an attacker from bypassing or misusing an update mechanism. Providing as little information as possible as part of the update mechanism can reduce the ability of an attacker to exploit it.

Refer to example in Table C.1 Annex C.

6.3.10 Software updates network interface

In the case where updates are delivered over a network interface, the device shall verify the authenticity and integrity of each update via a valid trust relationship.

NOTE:

1. Valid trust relationships include authenticated communication channels, presence on a network that requires the device to possess a critical security parameter or password to join, digital signature-based verification of the update, or confirmation by the user.

2. The validation of the trust relationship is essential to ensure that a non-authorised entity (e.g. device management platform or device) cannot install malicious code.

6.3.11 Software update risks mitigation

The manufacturer shall inform the user in a recognisable and apparent manner that a security update is required together with information on the risks mitigated by that update.

Refer to example in Table C.1 Annex C.

6.3.12 Software update disruption

The device shall notify the user when the application of a software update will disrupt the basic functioning of the device.

It can be critical for users that a device continues to operate during an update. Devices that fulfil a safety-relevant function are expected not to turn completely off in the case of an update where some minimal system functional capability is expected. Disruption to functionality can become a critical safety issue for some types of devices and systems if not considered or managed correctly.

Refer to example in Table C.1 Annex C.

NOTE: Notification can also be made by an associated service. This notification can include extra information, such as the approximate expected duration for which the device will be offline.

6.3.13 Software update support

The manufacturer shall publish the defined support period in an accessible way that is clear and transparent to the user.

6.3.14 Software update constraint

For constrained devices that cannot have their software updated due to the absence of software updates, the period and method of hardware replacement support and a defined support period should be published by the manufacturer in an accessible way that is clear and transparent to the user.

6.3.15 Isolable constrained devices product

For constrained devices that cannot have their software updated, the product should be isolable and the hardware replaceable.

There are some situations where devices cannot be patched. For constrained devices, a replacement plan should to be in place and be clearly communicated to the consumer. This plan would typically detail a schedule for when technologies will need to be replaced and, where applicable, when support for hardware and software ends.

6.3.16 Recognizable model designation

The model designation of the consumer IoT device shall be clearly recognizable, either by labelling on the device or via a physical interface.

This is often performed by communicating with a device over a logical interface. It can also be part of a UI.

Refer to example in Table C.1 Annex C.

Knowledge of the specific designation of the device is often required to check the defined support period of software updates or the availability of software updates.

6.4 Securing sensitive security parameters

6.4.1 Storing sensitive security parameters

Sensitive security parameters in persistent storage shall be stored securely by the device.

Secure storage mechanisms can be used to secure sensitive security parameters. Appropriate mechanisms include those provided by a Trusted Execution Environment (TEE), encrypted storage associated with the hardware, Secure Elements (SE) or Dedicated Security Components (DSC), and processing capabilities of software running on a Universal Integrated Circuit Card (UICC), according to ETSI TR 121 905, ETSI TS 102 221/embedded UICC according to GSMA SGP.22 Technical Specification v2.2.1.

NOTE: This requirement applies to persistent storage, but manufacturers can also implement similar approaches for sensitive security parameters in memory.

Refer to example in Table C.1 Annex C.

6.4.2 Hard-coded unique device identity

Where a hard-coded unique per device identity is used in a device for security purposes, it shall be implemented in such a way that it resists tampering by means such as physical, electrical or software.

Refer to example in Table B.

6.4.3 Hard-coded critical security parameters

Hard-coded critical security parameters in device software source code shall not be used.

Reverse engineering of devices and applications can easily discover credentials such as hard-coded usernames and passwords in software. These credentials can also be Application Programming Interfaces (API) keys that allow usage of security-sensitive functions in a remote service, or private keys used in the security of protocols that the device uses to communicate. Such credentials will often be found within a source code, which is a common bad practice. Simple obfuscation methods also used to obscure or encrypt this hard-coded information can be easily broken.

6.4.4 Unique critical security parameters

Any critical security parameters used for integrity and authenticity checks of software updates and for protection of communication with associated services in device software shall be unique per device and shall be produced with a mechanism that reduces the risk of automated attacks against classes of devices.

Refer to example in Table C.1 Annex C.

Provisioning a device with unique critical security parameters helps to protect the integrity and authenticity of software updates as well as the communication of the device with associated services. If global critical security parameters are used, their disclosure can enable wide-scale attacks on other IoT devices such as to enable the creation of botnets.

6.5 Communicate securely

6.5.1 Cryptography for secure communication

The consumer IoT device shall use best practice cryptography to communicate securely.

Appropriateness of security controls and the use of best practice cryptography is dependent on many factors including the usage context. As security is ever-evolving, it is difficult to give prescriptive advice about cryptography or other security measures without the risk of such advice quickly becoming obsolete.

6.5.2 Cryptography for consumer IoT device

The consumer IoT device should use reviewed or evaluated implementations to deliver network and security functionalities, particularly in the field of cryptography.

Reviews and evaluations can involve an independent internal or external entity.

Refer to example in Table C.1 Annex C.

6.5.3 Updateable cryptographic algorithms and primitives

Cryptographic algorithms and primitives should be updateable.

NOTE: This is also known as "crypto-agility".

For devices that cannot be updated, it is important that the intended lifetime of the device does not exceed the recommended usage lifetime of cryptographic algorithms used by the device (including key sizes).

6.5.4 Access to device functionality

Access to device functionality via a network interface in the initialized state should only be possible after authentication on that interface.

NOTE: Functionality can vary significantly on the use case and can encompass a range of things, including access to personal data and device actuators.

There are devices that provide public, open data for example in the Web of Things. These devices are accessible without authentication to provide open access to all.

The device can be compromised via vulnerabilities in network services. A suitable authentication mechanism can protect against unauthorized access and can contribute to defence-in-depth in the device.

6.5.5 Access to security-relevant configurations

Device functionality that allows security-relevant changes in configuration via a network interface shall only be accessible after authentication. The exception is for network service protocols that are relied upon by the device and where the manufacturer cannot guarantee what configuration will be required for the device to operate.

NOTE: Protocols that are an exception include Address Resolution Protocol (ARP), Dynamic Host Configuration Protocol (DHCP), Domain Name System (DNS), Internet Control Message Protocol (ICMP), and Network Time Protocol (NTP).

Refer to example in Table C.1 Annex C.

6.5.6 Encryption of critical security parameters

Critical security parameters should be encrypted in transit, with such encryption appropriate to the properties of the technology, risk and usage.

6.5.7 Protection of critical security parameters

The consumer IoT device shall protect the confidentiality of critical security parameters that are communicated via remotely accessible network interfaces.

Many different methods exist for enrolment and authentication. Some authentication values are provided by out-of-band authentication mechanisms, such as a QR code, and some are human-readable, such as a password.

Where an authentication mechanism uses unique values per authentication attempt (e.g. in a challenge-response mechanism or when using one-time passwords as a second factor), the response is not the authentication value itself. However, it is still good practice to apply confidentiality protection to those values.

Confidentiality protection can be achieved using an encrypted communication channel or payload encryption. This is often done using protocols or algorithms at least as strong as the key material transmitted, however other mitigations, such as the need for close proximity, are available.

6.5.8 Secure management process

The manufacturer shall follow secure management processes for critical security parameters that relate to the device.

The use of open, peer-reviewed standards for critical security parameters, commonly referred to as ‘key management’ is strongly encouraged.

6.6 Minimising attack surfaces

The principle of least privilege is a foundation stone of good security engineering, applicable to IoT as much as in any other field of application.

6.6.1 Unused network and logical interfaces

All unused network and logical interfaces shall be disabled.

Refer to example in Table C.1 Annex C.

6.6.2 Minimise unauthenticated disclosure

In the initialized state, the network interfaces of the device shall minimize the unauthenticated disclosure of security-relevant information.

Security-relevant information can be exposed over a network interface as part of the initialization process. When security-relevant information is shared by a device when establishing a connection, it can be used by attackers to identify vulnerable devices.

Refer to example in Table C.1 Annex C.

6.6.3 Exposure of physical interfaces

Device hardware should not unnecessarily expose physical interfaces to attack.

Physical interfaces can be used by an attacker to compromise firmware or memory on a device. Unnecessarily refers to the manufacturer's assessment of the benefits of an open interface, used for user functionality or for debugging purposes.

Refer to example in Table C.1 Annex C.

6.6.4 Existence of debug interface

Where a debug interface is physically accessible, it shall be disabled in software.

Refer to example in Table C.1 Annex C.

6.6.5 Enabling software services

The manufacturer should only enable software services that are used or required for the intended use or operation of the device.

Refer to example in Table C.1 Annex C.

6.6.6 Minimise code

Code should be minimized to the functionality necessary for the service or device to operate.

Refer to example in Table C.1 Annex C.

6.6.7 Running software

Software should run with the least necessary privileges, taking account of both security and functionality.

Refer to example in Table C.1 Annex C.

Software attacks on devices that aim to corrupt memory can be mitigated through mechanisms such as stack canaries and Address Space Layout Randomization (ASLR). The manufacturer can use platform security features where they are available to help further reduce the risk. Reducing privileges that they run at and minimizing code also helps to mitigate this risk.

6.6.8 Hardware-level access control mechanism

The device should include a hardware-level access control mechanism for memory.

Software exploits often use the lack of access control in memory to execute malicious code. Access control mechanisms limit whether data in memory on the device can be executed. Suitable mechanisms

include technologies such as Memory Management Unit (MMU) or Memory Protection Unit (MPU), executable space protection (e.g. NX bits), memory tagging, and trusted execution environments.

6.6.9 Development of software

The manufacturer should follow secure development processes for software deployed on the device.

Secure development processes, including using version control, or enabling security-related compiler options (e.g., stack protection) can help ensure software artefacts are more secure. Manufacturers can use these options when using toolchains that support them.

6.7 Ensure software integrity

6.7.1 Secure boot mechanisms

The consumer IoT device should verify its software using secure boot mechanisms.

A hardware root of trust is one way to provide strong attestation as part of a secure boot mechanism. A hardware root of trust is a component of a system from which all other components derive their 'trust' - i.e., the source of cryptographic trust within that system. To fulfil its function, the hardware root of trust must be reliable and resistant to both physical and logical tampering, as there is no mechanism to determine whether the component has failed or been altered. By utilizing a hardware root of trust, a device can have confidence in the results of cryptographic functions, such as those utilized for secure boot. A hardware root of trust can be either backed by mechanisms used for secure storage of credentials or other alternatives providing baseline levels of security assurance proportionate to the required level of security for a given device.

6.7.2 Unauthorized change

If an unauthorized change is detected to the software, the device should alert the user and/or administrator to the issue and should not connect to wider networks other than those necessary to perform the alerting function.

The ability to recover remotely from unauthorized changes can rely on a known good state, such as locally storing a known good version to enable safe recovery and updating of the device. This will avoid denial of service and costly recalls or maintenance visits, whilst managing the risk of potential takeover of the device by an attacker subverting update or other network communications mechanisms.

If a consumer IoT device detects an unauthorized change to its software, it should be able to inform the right stakeholder. In some cases, devices can have the ability to be in administration mode.

Refer to example in Table C.1 Annex C.

NOTE: An attack that forces a device to revert to a known good state can introduce a DoS risk if the device is unable to successfully perform this or if the attacker is able to repeatedly cause this effect.

6.8 Ensure secure personal data

6.8.1 Confidentiality of transiting data

The confidentiality of personal data transiting between a device and a service, especially associated services, should be protected, with best practice cryptography.

6.8.2 Confidentiality of sensitive data

The confidentiality of sensitive personal data communicated between the device and associated services shall be protected, with cryptography appropriate to the properties of the technology and usage.

NOTES:

1. In the context of this requirement, 'sensitive personal data' is data whose disclosure has a high potential to cause harm to the individual. What is to be treated as 'sensitive personal data' varies across products and use cases, but examples are: video stream of a home security camera, payment information, content of communication data and timestamped location data. Carrying out security and data protection impact assessments can help the manufacturer make appropriate choices.

2. Associated services in this context are typically cloud services. Moreover, these services are controlled or can be influenced by the manufacturer. These services typically are not operated by the user.

3. Confidentiality protection often includes integrity protection according to best practice cryptography.

6.8.3 External sensing capabilities

All external sensing capabilities of the device shall be documented in an accessible way that is clear and transparent for the user.

Refer to example in Table C.1 Annex C.

6.9 Systems resilient to outages

IoT devices and services should be designed to provide a level of resilience such that consumers are not significantly impacted when there are outages of data networks and power.

6.9.1 Building resilience

Resilience should be built into consumer IoT devices and services, taking into account the possibility of outages of data networks and power.

6.9.2 Remain functioning

Consumer IoT devices should remain operating and locally functional in the case of a loss of network access and should recover cleanly in the case of restoration of a loss of power.

NOTE: Recovering cleanly normally involves resuming connectivity and functionality in the same or improved state.

6.9.3 Connection to networks

The consumer IoT device should connect to networks in an expected, operational and stable state and in an orderly fashion, taking the capability of the infrastructure into consideration.

Refer to example in Table C.1 Annex C.

IoT systems and devices are relied upon by consumers for increasingly important use cases that can be safety-relevant or life-impacting. Keeping services running locally if there is a loss of network is one of the measures that can be taken to increase resilience. Other measures can include building redundancy into associated services as well as mitigations against Distributed Denial of Service (DDoS)

attacks or signalling storms, which can be caused by mass-reconnections of devices following an outage. It is expected that the level of resilience necessary is proportionate and determined by usage, with consideration given to others that rely on the system, service or device given that an outage can have a wider impact than expected.

Orderly reconnection means in a manner that takes explicit steps to avoid simultaneous requests, such as for software updates or reconnections, from a large number of IoT devices. Such explicit steps can include the introduction of a random delay before a reconnection attempt according to an incremental back-off mechanism.

6.10 Secure telemetry data

6.10.1 Examine for security anomalies

If telemetry data is collected from consumer IoT devices and services, such as usage and measurement data, it should be examined for security anomalies.

Examining telemetry, including log data, is useful for security evaluation and allows for unusual circumstances to be identified early and dealt with, minimizing security risk and allowing quick mitigation of problems.

Refer to example in Table C.1 Annex C.

6.11 Deleting user data

6.11.1 User-friendly deletion

The user shall be provided with functionality such that user data can be erased from the device in a simple manner.

NOTE: User data in this context means all individual data which is stored on the IoT device including personal data, user configuration and cryptographic material such as user passwords or keys.

6.11.2 Removal of personal data from associated services

The consumer should be provided with functionality on the device such that personal data can be removed from associated services in a simple manner. Such functionality is intended for situations when the consumer wishes to remove a service from the device or when there is a transfer of ownership of the device or when the consumer wishes to dispose of the device. It is expected that such functionality is compliant to applicable data protection law, including the General Data Protection Regulation (GDPR).

Removing personal data easily means that minimal steps are required to complete that action.

Such functionality can potentially present an attack vector.

6.11.3 Instruction to delete data

Users should be given clear instructions on how to delete their personal data.

6.11.4 Confirmation on data deletion

Users should be provided with clear confirmation that personal data has been deleted from services, devices and applications.

Consumer IoT devices often change ownership and will eventually be recycled or disposed of. Mechanisms should be provided that allow the consumer to remain in control and remove personal data from services, devices and applications. When a consumer wishes to completely remove their personal data, they also expect retrospective deletion of backup copies.

Deleting personal data from a device or service is often not simply achieved by resetting a device back to its factory default state. There are many use cases where the consumer is not the owner of a device but wishes to delete their own personal data from the device and all associated services such as cloud services or mobile applications.

Refer to example in Table C.1 Annex C.

6.12 Installation and maintenance of devices

6.12.1 User-friendly installation and maintenance

Installation and maintenance of consumer IoT devices should involve minimal decisions by the user and should follow security best practices on usability.

Refer to example in Table C.1 Annex C.

6.12.2 Users' guidance to set up device

The manufacturer should provide users with guidance on how to securely set up their device.

However, the ideal is for a process that involves the minimum of human intervention, and which achieves a secure configuration automatically.

6.12.3 Users' guidance to check device set up

The manufacturer should provide users with guidance on how to check whether their device is securely set up.

Security issues caused by consumer confusion or misconfiguration can be reduced and sometimes eliminated by properly addressing complexity and poor design in user interfaces. Clear guidance to users on how to configure devices securely can also reduce their exposure to threats.

Generally, the average overhead of securely setting up a device is higher than the average overhead of checking whether a device is securely setup. The check of a secure setup, from a process standpoint, can be undertaken in a large part by the manufacturer through an automated process that communicates with the device remotely. Part of such an automated process could include validation of the device's capacity to establish a secure communication channel.

6.13 Validate input data

6.13.1 Validate data input

The consumer IoT device software shall validate data input via user interfaces or transferred via APIs or between networks in services and devices.

Systems can be subverted by incorrectly formatted data or code transferred across different types of interfaces. Automated tools such as fuzzers can be used by attackers or testers to exploit potential gaps and weaknesses that emerge as a result of not validating data.

Refer to example in Table C.1 Annex C.

7. Data protection requirements for consumer IoT

Many consumer IoT devices process personal data. It is expected that manufacturers provide features within consumer IoT devices that support the protection of such personal data. In addition, there exist laws and regulations, which are the Personal Data Protection Act 2010 (Laws of Malaysia Act 709) (PDPA), Guidelines for Secure Internet of Things (IoT), Information and Network Security - Malaysia Critical Security Controls (MYCSC) and The Personal Data Protection Code of Practice for the Communications Class Data Users 2017 that relate to the protection of personal data in consumer IoT devices.

According to the Security Principle in Act 709 , a data user shall, when processing personal data, take practical steps to protect the personal data from any loss, misuse, modification, unauthorized or accidental access or disclosure, alteration, or destruction. This Technical Code intends to help manufacturers of consumer IoT devices to provide a number of features for the protection of personal data from a strictly technical perspective.

a) The manufacturer shall provide consumers with clear and transparent information about what personal data is processed, how it is being used, by whom, and for what purposes, for each device and service. This also applies to third parties that can be involved, including advertisers.

b) Where personal data is processed on the basis of consumers' consent, this consent shall be obtained in a valid way.

   Obtaining consent ‘in a valid way’ normally involves giving consumers a free, obvious and explicit opt-in choice of whether their personal data can be used for a specified purpose.

c) Consumers who have given consent for the processing of their personal data shall have the capability to withdraw it at any time.

   Consumers expect to be able to preserve their privacy by configuring IoT device and service functionality appropriately.

d) If telemetry data is collected from consumer IoT devices and services, the processing of personal data should be kept to the minimum necessary for the intended functionality.

e) If telemetry data is collected from consumer IoT devices and services, consumers shall be provided with information on what telemetry data is collected, how it is being used, by whom, and for what purposes.

f) Manufacturers shall provide consumers with the latest and adequate anti-virus software to avoid putting the personal data of consumers at risk consequent to virus infections and other malware.

g) Manufacturers shall ensure that any backup data is sanitised or deleted according to the data retention policy in PDPA or data retention laws for regulated industries if exist.

Annex A
(informative)

Abbreviations

API	Application Programming Interface
ARP	Address Resolution Protocol
ASLR	Address Space Layout Randomization
BLE	Bluetooth Low Energy
CVD	Coordinated Vulnerability Disclosure
DDoS	Distributed Denial of Service
DFU	Direct Firmware Update
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
DoS	Denial-of-Service
DSC	Dedicated Security Components
DTS	Device Telemetry Service
DVAD	Digital Voice Assistant Device
GDPR	General Data Protection Regulation
GSM	Global System for Mobile communications
GSMA	GSM Association
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
ICMP	Internet Control Message Protocol
IEEE	Institute of Electrical and Electronics Engineers
IoT	Internet of Things
IP	Internet Protocol
JTAG	Joint Test Action Group
LAN	Local Area Network
LTE-m	Long Term Evolution Machine
LoRaWAN	Long Range Wide Area Network
MAC	Media Access Control
MMU	Memory Management Unit
MPU	Memory Protection Unit

MYCSC	Malaysia Critical Security Controls
NIST	National Institute of Standards and Technology
NTP	Network Time Protocol
NX	No execute
OTP	One-Time Password
PAN	Personal Area Network
PDPA	Personal Data Protection Act
PIN	Personal Identification Number
QR	Quick Response
SBOM	Software Bill of Materials
SDO	Standards Development Organisation
SE	Secure Elements
SSID	Service Set Identifier
SWD	Serial Wire Debug
TEE	Trusted Execution Environment
TLS	Transport Layer Security
TS	Technical Specification
TV	Television
UART	Universal Asynchronous Receiver-Transmitter
UI	User Interface
UICC	Universal Integrated Circuit Card
USB	Universal Serial Bus
WAN	Wide Area Network
WiFi	Wireless Fidelity

Annex B
(normative)

Terms and Definitions

Table B.1 Examples related to Terms and Definitions

No.	Term	Detail	Example
1	Associated services	Refer to 4.2	1. Associated services can include mobile applications, cloud computing or storage, and third-party Application Programming Interfaces (APIs). 2. A device transmits telemetry data to a third-party service chosen by the device manufacturer. This service is an associated service.
2	Authentication mechanism	Refer to 4.3	An authentication mechanism can be the request of a password, the scanning of a QR code, or the use of a biometric fingerprint scanner.
3	Authentication value	Refer to 4.4	When the authentication mechanism is to request a password, the authentication value can be a character string. When the authentication mechanism is biometric fingerprint recognition, the authentication value can be the index fingerprint of the left hand.
4	Best practice cryptography	Refer to 4.5	The device manufacturer uses a communication protocol and cryptographic library provided with the IoT platform and where that library and protocol have been assessed against feasible attacks, such as replay.
5	Constrained device	Refer to 4.6 (NOTE 1)	1. A window sensor's battery cannot be charged or changed by the user; this is a constrained device. 2. The device cannot have its software updated due to storage limitations, resulting in hardware replacement or network isolation being the only options to manage a security vulnerability. 3. A low-powered device uses a battery to enable it to be deployed in various locations. Performing high-power cryptographic operations would quickly reduce the battery life, so it relies on a base station or hub to perform validations on updates. 4. The device has no display screen to validate binding codes for Bluetooth pairing. 5. The device has no ability to input, such as via a keyboard, authentication information.
6	Constrained Devices	Refer to 4.6 (NOTE 2)	A device is mains powered and communicates primarily using TLS (Transport Layer Security).

Table B.1 Examples related to Terms and Definitions *(continued)*

No.	Term	Detail	Example
7	Critical security parameter	Refer to 4.9	Secret cryptographic keys, authentication values such as passwords, PINs, private components of certificates.
8	Debug interface	Refer to 4.10	Test points, UART, SWD, JTAG.
9	Isolable	Refer to 4.17	A smart fridge has a touchscreen-based interface that is network-connected. This interface can be removed without stopping the fridge from keeping the contents chilled.
10	Physical interface	Refer to 4.23	Radios, ethernet ports, serial interfaces such as USB, and those used for debugging.
11	Public security parameter	Refer to 4.24	1. A public key to verify the authenticity or integrity of software updates. 2. Public components of certificates.
12	Security module	Refer to 4.26	A device contains a hardware root of trust, a cryptographic software library that operates within a trusted execution environment, and software within the operating system that enforces security such as user separation and the update mechanism. All of these make up the security module.
13	Software service	Refer to 4.29	A runtime for the programming language used within the device software or a daemon that exposes an API used by the device software, e.g., a cryptographic module's API.
14	Telemetry	Refer to 4.30	A consumer IoT device reports software malfunctions to the manufacturer enabling them to identify and remedy the cause.

Annex C
(normative)

Cyber Security Requirements for Consumer IoT

Table C.1 Examples of implementation of Cyber Security Requirements for Consumer IoT

No.	Requirement	Detail	Example
1	No universal default password	Refer to 6.1(a)	During initialization a device generates certificates that are used to authenticate a user to the device via an associated service like a mobile application.
2	No universal default password	Refer to 6.1(b)	Pre-installed passwords are sufficiently randomized.
3	No universal default password	Refer to 6.1(d)	1. For biometric authentication values, the device manufacturer allows this change in authentication value through retraining against a new biometric. 2. A parent in a household creates an account on the device for their child and selects and manages the PIN or password that the child uses. The parent is an administrator on the device and can restrict the child from changing the PIN or password. 3. To make it simple for the user to change a password, the manufacturer designs the password change process in a way that it requires a minimal number of steps. The manufacturer explains the process in a user manual and in a video tutorial.
4	No universal default password	Refer to 6.1(e)	1. A device has a limitation on the number of authentication attempts within a certain time interval. It also uses increasing time intervals between attempts. 2. The client application is able to lock an account or to delay additional authentication attempts after a limited number of failed authentication attempts.
5	Keep software updated	Refer to 6.3.1	1. The first stage boot loader on a device is written once to device storage and from then on is immutable. 2. On devices with several microcontrollers (e.g., one for communication and one for the application) some of them might not be updateable.

Table C.1 Examples of the implementation of Cyber Security Requirements for Consumer IoT
(continued)

No.	Provision	Detail	Example
6	Keep software updated	Refer to 6.3.2	1. Measures can include the use of authentic software update servers, integrity protected communications channels, verifying the authenticity and integrity of software updates. It is recognized that there are great variances in software update mechanisms and what constitutes "installation". 2. An anti-rollback policy based on version checking can be used to prevent downgrade attacks.
7	Keep software updated	Refer to 6.3.4	Under the EU Product Legislation, a feature update could change the intended use of a device and thus turn it into a new product, requiring a new conformity assessment to be conducted. However, a software update with limited impact could be considered a maintenance update which would not require a new conformity assessment. More information on the impact of software updates in the context of the EU Product Legislation can be found in the Blue Guide.
8	Keep software updated	Refer to 6.3.5	1. The user could be shown the existence of updates via the interface with which the device is initialized. 2. A device checks for available updates daily at a randomized time.
9	Keep software updated	Refer to 6.3.8	A particular software update involves a third-party vendor of software libraries, an IoT device manufacturer, and an IoT service platform operator. Collaboration between these stakeholders ensures appropriate timeliness of the software update.
10	Keep software updated	Refer to 6.3.9	When a device detects that an update could not be delivered or applied successfully (by failing integrity or authentication checks), the device can mitigate information leakage by not providing any information about the failure to the initiator of the update process. However, the device can generate a log entry and deliver notification of the log entry to a trusted peer (e.g., a device administrator) over a secure channel, so that the occurrence of the incident is known, and the owner or administrator of the device can make an appropriate response.
11	Keep software updated	Refer to 6.3.11	The manufacturer informs the user that an update is required via a notification on the user interface or via an email.

Table C.1 Examples of the implementation of Cyber Security Requirements for Consumer IoT
(continued)

No.	Provision	Detail	Example
12	Keep software updated	Refer to 6.3.12	1. A notification includes information about the urgency and approximate expected duration of downtime. 2. During an update, a watch will continue to display the time, a home thermostat will continue to maintain a reasonable temperature and a smart lock will continue to lock and unlock a door.
13	Keep software updated	Refer to 6.3.16	A device has a Hypertext Transfer Protocol (HTTP) (or Hypertext Transfer Protocol Secure (HTTPS) when appropriate) API that reports the model designation (after user authentication).
14	Securely store sensitive security parameters	Refer to 6.4.1	1. The root keys involved in authorization and access to licensed radio frequencies (e.g., Long Term Evolution Machine (LTE-m) cellular access) are stored in a UICC. 2. A remote-controlled door-lock using a Trusted Execution Environment (TEE) to store and access the sensitive security parameters. 3. A wireless thermostat stores the credentials for the wireless network in a tamper protected microcontroller rather than in external flash storage.
15	Securely store sensitive security parameters	Refer to 6.4.2	A master key used for network access that is unique to the device is stored in UICC which is compliant to relevant ETSI standards (see, for example ETSI TS 102 221).
16	Securely store sensitive security parameters	Refer to 6.4.4	1. A different symmetric key is deployed on every device of the same product class for generating and verifying message authentication codes for software updates. 2. The device uses the manufacturer's public key to verify a software update. This is not a critical security parameter and does not need to be unique per device.
17	Communicate securely	Refer to 6.5.2	Distributed software libraries within the development and test community, certified software modules, and hardware equipment crypto-service providers (such as the Secure Element and Trust Execution Environment) are all reviewed or evaluated.
18	Communicate securely	Refer to 6.5.5	Security-relevant changes include permission management, configuration of network keys and password changes.

Table C.1 Examples of the implementation of Cyber Security Requirements for Consumer IoT
(continued)

No.	Provision	Detail	Example
19	Minimize exposed attack surfaces	Refer to 6.6.1	1. An administrative UI that is supposed to be accessed from the LAN is not accessible from the WAN by default. 2. A Direct Firmware Update (DFU) service exposed over Bluetooth® Low Energy is used for development but not expected to be used in production. It is disabled in the final product.
20	Minimize exposed attack surfaces	Refer to 6.6.2	When finding vulnerable devices throughout the whole IP address space, security-relevant information could be information about the device configuration, kernel version or software version.
21	Minimize exposed attack surfaces	Refer to 6.6.3	1. A micro-USB port meant to be used to power the device only if physically configured so as not to also allow command or debug operations. 2. A router or swtch port that are not in use should be put in shut down mode.
22	Minimize exposed attack surfaces	Refer to 6.6.4	A UART serial interface is disabled through the bootloader software on the device. No logon prompt and no interactive menu is available due to this disabling.
23	Minimize exposed attack surfaces	Refer to 6.6.5	The manufacturer does not provision the device with any background processes, kernel extensions, commands, programs or tools that are not required for the intended use.
24	Minimize exposed attack surfaces	Refer to 6.6.6	"Dead" or unused code is removed and not considered to be benign.
25	Minimize exposed attack surfaces	Refer to 6.6.7	1. Minimal daemons or processes run with "root" privileges. In particular the processes that use network interfaces require unprivileged users rather than requiring a "root" user. 2. Applications running on a device that includes a multi-user operating system (e.g., Linux) use different users for each component or service.
26	Ensure software integrity	Refer to 6.7.2	A thermostat in a room can have a user mode; this mode prevents changing of other settings. If an unauthorized change to software is detected, an alert to the administrator is appropriate, as the administrator has the ability to act on the alert (whereas a user does not).

Table C.1 Examples of the implementation of Cyber Security Requirements for Consumer IoT
(concluded)

No.	Provision	Detail	Example
27	Ensure personal data is secure	Refer to 6.8.3	An external sensing capability can be an optic or acoustic sensor.
28	Make systems resilient to outages	Refer to 6.9.3	1. A smart home loses connection to the internet following a power outage. When the network connection is restored, the devices in the home reconnect after a randomized delay to minimize network utilization. 2. After making an update available, the manufacturer notifies devices in batches to prevent them all simultaneously downloading the update.
29	Information obtained from telemetry data	Refer to 6.10.1	1. Security anomalies can be represented by a deviation from normal behaviour of the device, as expressed by the monitored indicators, for example an abnormal increase of failed login attempts. 2. Telemetry from multiple devices allows a manufacturer to notice that updates are failing due to invalid software update authenticity checks.
30	Deletion of personal data by factory reset	Refer to 6.11.4	A user can have temporary usage of a consumer IoT product within a rented apartment. Carrying out a factory reset of the product can remove configuration settings or disable the device to the detriment of the apartment owner and a future user. A factory reset, deleting all data from the IoT device, would not be an appropriate way to delete the personal data of a single user in a shared use context such as this.
31	Make installation and maintenance of devices easy	Refer to 6.12.1	The user uses a wizard to setup the device where a subset of configuration options is presented with the common defaults already specified and with appropriate security options already turned on by default.
32	Validate input data	Refer to 6.13.1	1. The device receives data that is not of the expected type, for example executable code rather than user inputted text. The software on the device has been written so that the input is parameterized or "escaped", preventing this code from being run. 2. Out of range data is received by a temperature sensor. The device identifies the data as being outside of the possible bounds and discards it, and the event is captured in telemetry.

Bibliography

[1] ISO/IEC 29147: *Information technology - Security techniques - Vulnerability Disclosure*

[2] ETSI TS 102 221: Smart Cards; UICC-Terminal interface; Physical and logical characteristics

[3] ETSI TR 103 621 v1.1.1: *Guide to Cyber security for consumer Internet of Things*

[4] ETSI EN 103 645 v2.1.2: *CYBER; Cyber Security for Consumer Internet of Things: Baseline Requirements*

[5] ETSI TS 103 701 v1.1.1*: CYBER; Cyber Security for Consumer Internet of Things: Conformance Assessment of Baseline Requirements*

[6] ETSI TR 121 905: *Digital cellular telecommunications system (Phase 2+) (GSM); Universal Mobile Telecommunications System (UMTS); LTE; Vocabulary for 3GPP Specifications (3GPP TR 21.905)*

[7] GSMA: *Coordinated Vulnerability Disclosure (CVD) Programme*

[8] GSMA: *GSMA IoT Security Guidelines and Assessment*

[9] GSMA: *SGP.22 Technical Specification v2.2.1*

[10] DIN SPEC 27072: *Information Technology - IoT Capable Devices - Minimum Requirements for Information Security*

[11] IMDA: *Internet of Things (IoT) Cyber Security Guide*

[12] NIST IoT Device Security

[13] NIST Special Publication 800-63B*: Digital Identity Guidelines - Authentication and Lifecycle Management*

[14] Regulation (EU) 2016/679 of the European Parliament and of the Council: *the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation)*

[15] RFC 7228: *Terminology for Constrained-Node Networks*

[16] OASIS: *CSAF Common Vulnerability Reporting Framework (CVRF)*

[17] Commission Notice: *The "Blue Guide" on the Implementation of EU Products Rules 2016 (Text with EEA relevance), 2016/C 272/01*

[18] IoT Security Foundation: *Understanding the Contemporary Use of Vulnerability Disclosure in Consumer Internet of Things Product Companies*

[19] IoT Security Foundation: *Vulnerability Disclosure - Best Practice Guidelines*

[20] F-Secure: *IoT threats: Explosion Of 'Smart' Devices Filling Up Homes Leads to Increasing Risks*

[21] W3C: *Web of Things at W3C*

Acknowledgements

Members of the Internet of Things (IoT) and Smart Sustainable Cities Working Group

Dr Gopinath Rao Sinniah (Chairman/Draft Lead)	Favoriot Sdn Bhd
Mr Mohd Zakir Hussin Baharuddin (Vice Chairman)	Telekom Malaysia Berhad
Associate Professor Ir Dr Yusnani Mohd Yussoff (Draft Lead)	Universiti Teknologi Mara
Mr Mohamad Norzamir Mat Taib (Secretariat)	Malaysian Technical Standards Forum Bhd
Mr Ang Kah Heng	Cyberview Sdn Bhd
Mr Meng Keen Wai/ Mr Wong Chup Woh	Maxis Broadband Sdn Bhd
Professor Lau Sian Lun	Sunway University College Sdn Bhd
Dr Teng Kah Hou	UCSI Education Sdn Bhd
Dr Mohd Yamani Idna Idris	Universiti Malaya
Professor Dr Borhanuddin Mohd Ali	Universiti Putra Malaysia
Dr Hafizal Mohamad	Universiti Sains Islam Malaysia
Ms Alisa Rafiqah Adenan	Universiti Teknologi Mara

By invitation

Ms Nur Amilin Mohd Khazani	PLANMalaysia