Cybersecurity Labelling Scheme for IoT
Publication No. 4

Assessment Methodology

October 2024
Version 1.1

FOREWORD

The Cybersecurity Labelling Scheme for IoT [CLS(IoT)] is part of Cyber Security Agency's (CSA) efforts to better secure Singapore's cyberspace and to raise cyber hygiene levels. It aims to improve security awareness by making security provisions more transparent to consumers and empowers consumers to make informed purchasing decisions for products with better security using the information on the cybersecurity label.

Under the CLS(IoT), the cybersecurity label provides an indication of the level of security in the network-connected smart devices.

The CLS(IoT) seeks to incentivise developer/manufacturers to develop and provide products with enhanced cybersecurity provisions. The labels also serve to differentiate smart devices with better cybersecurity safeguards in the market, from their competitors.

At the same time, CSA intends to engage other like-minded partners for mutual recognition of the CLS(IoT) with the objective of eliminating duplicated assessments across national boundaries.

The CLS(IoT) is an initiative under the Safer Cyberspace Masterplan, to create a safer cyberspace and protect the public and enterprises against cyber threats, as Singapore moves towards a Digital Economy and Smart Nation.

The CLS(IoT) is owned and managed by the Cybersecurity Certification Centre (CCC), under the ambit of the Cyber Security Agency of Singapore (CSA).

AMENDMENT RECORD

Version	Date	Author	Changes
1.0	September 2023	Cyber Security Agency of Singapore	Released
1.1	October 2024	Cyber Security Agency of Singapore	Updated Requirements

CONTENTS

INTRODUCTION

This document specifies the assessment methodology for the Cybersecurity Labelling Scheme for IoT [CLS(IoT)].

INTENDED USAGE

The assessment methodology is developed to provide clarification to better align expectations and requirements for each security provision to facilitate easier adoption of the CLS(IoT).

This is done through specifying three main components for each security provision:

1. **Minimum requirements** – Specifies what is required of the IoT device to fulfil the provision.

2. **Supporting evidence** – Provides examples and/or suggestions of expected evidence that should be provided by the developer to allow the assessor to determine if a particular provision is fulfilled.

3. **Assessment** – Specifies what the assessor will check/examine from the supporting evidence provided to determine if the minimum requirement specified for each provision is met.

Disclaimer:

While this document provides examples of acceptable supporting evidence for each security provision, it is by no means exhaustive, and the required supporting evidence may deviate from what is stated.

The CCC reserves the right to request for further clarification, more supporting evidence and reject any evidence if what is provided is deemed insufficient in fulfilling the security provision's requirements.

The evidence that developers provide to fulfil the minimum requirements for security clauses shall not contain personal identifiable information (PII).

PROVISIONS FOR EACH CLS(IOT) LEVEL

Table 1 details the mandatory provisions per CLS(IoT) level.

CLS(IoT) Levels	Assessment Activities	Format	Mandatory Provisions
Level 1	Security Baseline Requirements	Developer's declaration of conformity.	5.1, 5.2-1, 5.3-2, 5.3-3, 5.3-7, 5.3-8, 5.3-10, 5.3-13, 5.3-16
Level 2	Adherence to International Standards		Level 1 provisions, 5.4, 5.5-1, 5.5-5, 5.5-7, 5.5-8, 5.6-1, 5.6-2, 5.6-4, 5.8-2, 5.8-3, 5.11-1, 5.13-1, 6.1, 6.2, 6.3, 6.5
Level 3	Lifecycle Requirements and Software Binary Analysis	Lifecycle Requirements based on Developer's declaration of conformity. Software Binary Analysis based on independent assessment by testing laboratory.	Level 1 provisions, level 2 provisions, and CK-LP
Level 4	Penetration Testing	Testing laboratory's report summarising the tests performed and the results.	

Table 1 - Mandatory Provisions for each CLS(IoT) Level

DEFINITIONS

Term	Definition
Authentication Interface	Interfaces on the device (or its companion application/services) that requires user interaction for authentication. Examples: GUI login portal, Mobile application login page, etc.
Authentication Mechanisms	Credential that is utilised by the user to authenticate themselves to the device using an authentication interface. Examples: passwords, tokens, smart cards, digital signatures, biometrics, etc.
Contact Mechanisms	Ways or avenues that the user can contact the developer, generally this is for the purpose of vulnerability reporting. Examples: Web form, contact emails, hotlines, etc.
Update Mechanisms	Ways that an IoT device can receive and install firmware updates. Examples: Automatic update and manual update feature found on the device.
Automatic Updates	Firmware updates are downloaded and installed on the device without the need for user interaction.
Manual Updates	Firmware updates that require user interaction. Examples: Pressing update button on GUI/application, downloading update from developer website, and installing it on device, etc.
Constrained Devices	Majority of the devices that constitute IoTs are known as constrained devices. They have limited CPU, memory, and power resources. Examples: sensors, smart objects, or smart devices, etc.
Trust relationships	A logical connection that is established between the device and another entity (e.g., update server, another device, mobile application, etc.) after the authentication process has passed.
Sensitive Security Parameters	These are parameters that are used to authenticate users with the device's interfaces, typically allowing the user to perform administrative actions that if abused, could be detrimental. Examples: Admin password, Wi-Fi password (SSID), device's private key for client authentication, root key used to encrypt other sensitive parameters, digital signature public key, etc.
Hard-coded	Embedding data directly into the source code of a program or firmware. Examples: hard-coded unique per device identifiers, hard-coded critical security parameters, etc.
Unique Per Device Identifiers	Usually, a numeric or alphanumeric code that identifies the specific version or model of a device. Example: product serial numbers, MAC addresses, device IDs, etc.
Critical Security Parameters	Critical security parameters used for integrity and authenticity checks of software updates shall be unique per device. Example: secret keys, private components of certificates, etc.
Security Problem	Any unmitigated risk or vulnerability in the device that threat actors can used to do damage to the user or device itself.

5.1 – NO UNIVERSAL DEFAULT PASSWORDS

Provision 5.1-1

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.

Minimum Requirements

- Factory pre-loaded passwords/PINs used for authentication interfaces (e.g., Telnet, FTP, SSH, device administrator portal, mobile application, etc.) shall be unique per device, where different units of the same model have different factory pre-loaded passwords/PINs.

- If the factory pre-loaded password/PINs are not unique, the device shall force the user to define a new password/PIN upon initialisation. The device shall not enter the operationalised state before the factory pre-loaded password/PIN is changed.

- If the device does not have pre-loaded password/PINs, the device shall remain in a disabled (non-functioning) state until the user defines a password/PIN.

Supporting Evidence

The developer shall list all the device's authentication mechanisms (e.g., Telnet, FTP, SSH via ethernet port, device administrator portal, mobile application, Bluetooth, etc.).

The developer shall also provide the NMAP scan to show all open ports (for LAN and WAN interfaces where applicable) and associated services on the device.

  - Port scan can be performed using nmap -sT -sU -A -p- <ip address>

The developer shall provide evidence (e.g., user manuals, screenshots, videos, etc.) to show or describe the device's setup process and guidance provided for (e.g., account registration/creation, adding or pairing of devices, etc.) a user to complete the initialisation of the device.

Assessment

The assessor shall examine the NMAP scan (for both LAN and WAN interfaces, where applicable) to determine if the list of authentication mechanisms provided by the developer is accurate.

For devices with non-unique pre-loaded passwords/PINs, the assessor shall check that users are required to define a new password/PIN upon device initialisation.

Provision 5.1-2

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.

(This provision is not applicable for cases whereby users are required to define password upon device initialisation.)

Minimum Requirements

- Pre-loaded passwords shall not have incremental counters (e.g., “password1”, “password2”, etc.).

- Pre-loaded passwords shall be sufficiently randomised using a random function.

- Pre-loaded passwords shall not be relatable in an obvious manner to publicly available information (e.g., Wi-Fi SSID, MAC address and product serial number, etc.).

- Pre-loaded passwords shall not have common strings or patterns (e.g., “Password123”, “QWERTY”, etc.).

Supporting Evidence

The developer shall describe how the pre-loaded passwords are generated (e.g., generated off-device, provisioned onto the device subsequently, during device’s initial boot-up sequence, etc.).

The developer shall declare the password generation method(s) used to randomise pre-loaded passwords (e.g., cryptographically secure pseudo random number generator, etc.).

The developer shall provide 10 instances of randomised pre-loaded passwords, in the form of screenshots/pictures generated using the password generation method declared earlier.

Assessment

The assessor shall check that the password generation method(s) used is able to sufficiently randomise generated pre-loaded passwords.

Additionally, the assessor shall examine the 10 instances of randomised pre-loaded passwords provided and determine if it is relatable in an obvious matter to publicly available information (e.g., Wi-Fi SSID, MAC address and product serial number, etc.).

Provision 5.1-3

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

Minimum Requirements

- For all mechanisms of the device where credentials are required to be transmitted shall be performed over a secure channel. Acceptable examples include but not limited to:
    - TLS 1.2 or higher, with acceptable cipher suites (refer to NIST SP 800-52),
    - For devices that use Bluetooth or Bluetooth Low Energy (BLE), Security Mode 1 with Security Level 3 or higher (excluding Security Mode 2 with Security Level 1),

- HTTP shall not be used to access to device's features (i.e., webGUI, mobile application, etc.).
    - Any attempts at accessing it through HTTP shall be redirected to HTTPS, OR
    - Any attempts at accessing it through HTTP shall be denied

- All credentials shall be securely stored on the device along with other associated services.
    - Passwords/tokens stored on the device shall be hashed using an approved hash algorithm, in accordance with NIST SP 800-131A.

Supporting Evidence

The developer shall provide a diagram including the entities (e.g., device, cloud, hub, etc.) and illustrating the secure communication protocol used. For examples, please refer to Annex A.

The developer shall provide evidence (e.g., Wireshark screenshot, etc.) showing the communication (transmission of credentials) between the entities. Where applicable, the IP addresses should be clearly labelled and within the appropriate subnets. Examples of such communication channels are, but not limited to:

- Mobile application to smart hub,
- Smart hub to device,
- Mobile application to device,
- Mobile application to cloud,
- Device to cloud.

If applicable, the developer shall provide evidence (e.g., video, screenshots, etc.) to show the redirection to HTTPS or denial of access when attempts are made to access the device's administration portal/GUI through HTTP.

If credentials are stored on the device, the developer shall describe how they are securely stored on it.

Additionally, the developer shall declare the hashing algorithm used for stored passwords and provide a screenshot of the stored hashed password (e.g., /etc/shadow).

Assessment

The assessor shall examine the diagram provided to ensure that credentials are transmitted over a secure channel.

The assessor shall check that the redirection to HTTPS and denial of access occurs when attempts are made to access the device's administration portal through HTTP.

Additionally, if credentials are stored on the device, the assessor shall check if they are securely stored.

For passwords stored on the device, the assessor shall check that the hashing algorithm used is in accordance with NIST SP 800-131A.

Provision 5.1-4

When 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.

Minimum Requirements

- For residential gateways (e.g., Wi-Fi Routers, Smart Hubs), password changes shall confirm to password requirements defined in IMDA TS RG-SEC[1].

- For devices with a GUI, the process of changing authentication values (e.g., PINs, passwords, etc.) shall be easy to execute.

- For devices without a GUI, to cater to users without technical knowledge, detailed instructions shall be provided to users (i.e., with the use of user manuals, instructional videos, etc.) to guide them on the process of changing authentication values.

Supporting Evidence

The developer shall provide evidence (e.g., screenshots, user manuals, videos, etc.) detailing how the user can change authentication values (e.g., reset password, change PIN, etc.) for all device platforms (e.g., device administration portal, mobile application, etc.) where authentication values can be changed.

For cases where the authentication value is a password, the developer shall also provide a screenshot of the failure message shown when the user attempts to set a password that does not meet minimum requirements (e.g., password length is too short, password does not meet complexity requirements, etc.).

Assessment

The assessor shall check if the device's GUI allows users to change authentication values.

For devices without a GUI, the assessor shall check if there are instructions provided to the user explaining how to change the device's authentication values.

If the authentication value is a password, the assessor shall examine the evidence provided to ensure that the interface will reject password changes that do not meet the minimum password requirements.

[1] Technical Specification on Security Requirements for Residential Gateways, IMDA

Provision 5.1-5

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.

Minimum Requirements

- Each authentication interface (e.g., device administrative portal, mobile application login, SSH, etc.) shall employ a brute-force attack prevention measure. Examples of typical brute-force prevention measures are, but not limited to:
    - Rate limiting policies that limit the number of authentications within an interval (e.g., locks/delays enforced after a threshold is met, etc.).
    - Multi-factor authentication (MFA) after initial setup.
    - One-Time-PINs/Passwords (OTPs).

- For devices that utilise rate limiting policies as a brute-force attack prevention measure:
    - For implementations where a delay is enforced after a threshold is met, it shall require **at least 100 days** to compromise via a brute-force attack.
    - For implementations where a rate limiting policy is employed by means of IP blocking, the chance of a brute-force attack being successful shall be **lower than 1%**.

Supporting Evidence

The developer shall describe the brute-force prevention measure implemented on the device's authentication mechanisms.

For brute-force attack prevention measures that are not rate limiting policies, the developer shall provide evidence in the form of screenshots of their validity periods (e.g., screenshot of the message on the device interface/mobile application/email showing how long the OTP is valid for).

For rate limiting policies, the developer shall declare the maximum number of attempts (threshold) within a given period (or attempts per IP address) and the result of reaching the maximum number of attempts (e.g., explain what is being locked/delayed).

Next, the developer shall provide evidence (e.g., videos, screenshots, etc.) showing the rate limiting policy in effect. Acceptable examples are, but not limited to:

- Error messages showing maximum login attempts has been reached.
- Error messages showing the locked-out period.

Lastly, the developer shall perform the calculation using the formula indicated below to show that the rate limiting policy employed to mitigate brute-force attacks meet the minimum requirements (stated above).

Number of days required for password cracking:

$$Number\ of\ Days\ required = \frac{Password\ Character\ Pool^{(Password\ Length)}}{Number\ of\ tries\ in\ 24hrs\ \times\ 2}$$

For IP blocking applications, to find out the chance of a brute-force attack being successful:

$$\%\ chance\ of\ brute - force\ attack\ succeeding =$$

$$\frac{Total\ number\ of\ IP\ address\ X\ Number\ of\ tries\ per\ IP\ address}{Password\ Character\ Pool^{(Password\ Length)}}\ X\ 100\%$$

Assessment

The assessor shall examine the calculations provided to ensure that the rate limiting policy employed on the device is sufficient in protecting the device against brute-force attacks by meeting the criteria of **requiring at least 100 days to compromise** or **a chance of brute-force attack succeeding below 1%.**

The assessor shall check if the error message shown to users on failed authentication attempts provide sufficient information on the brute-force mitigation measure (rate limiting policy, IP blocking, etc.).

5.2 – IMPLEMENT A MEANS TO MANAGE REPORTS OF VULNERABILITIES

Provision 5.2-1

The manufacturer shall make a vulnerability disclosure policy publicly available.

Minimum Requirements

- A Vulnerability Disclosure Policy (VDP) shall be indicated on the developer/manufacturer's website, minimally with the following information:
    - Contact information for reporting of issues,
    - Procedures regarding initial acknowledgement of receipt (e.g., automated email reply within 3 working days, etc.),
    - Procedures regarding status updates until resolution of vulnerability. Updates can be in the form of:
        - Declaration of affected devices by the developer,
        - Guidelines for users affected by the security vulnerability to follow,
        - Announcements to users to inform that the security team is working on a solution to address the vulnerability, etc.

Recommendation

For more information on proper VDP practices, refer to:

- OWASP: "Vulnerability Disclosure Cheat Sheet"
- OASIS: "CSAF Common Vulnerability Reporting Framework (CVRF)"
- ISO/IEC 29147:2018: "Information technology – Security Techniques – Vulnerability Disclosure"

Supporting Evidence

The developer shall provide a list of all avenues available to users for reporting vulnerabilities. Examples of such avenues include, but are not limited to:

- Contact numbers and/or email addresses on developer's website,
- User guidance documents that show contact numbers and/or email address,
- URL of the web form / website where users can report issues,
- Use of a vulnerability coordination and bug bounty platform (e.g., HackerOne, BugCrowd, Open Bug Bounty, etc.).

The developer shall provide the URL and a screenshot/PDF of their VDP webpage.

The developer shall also ensure that the procedures around the initial acknowledgement of receipt and status updates of vulnerability resolution is indicated within the VDP.

Assessment

The assessor shall check if the provided URL to the VDP is valid and if the screenshot/PDF accurately reflects the VDP webpage.

If no instruction is provided to the users on how to access the VDP webpage of the device, the assessor shall check if it is easily accessible through the developer's website or the device's product page.

If the developer uses a bug bounty platform, the assessor shall check that there is a URL on the VDP webpage that directs users to the bug bounty platform.

The assessor shall check that the VDP provides a contact mechanism (i.e., web form, contact email, hotline, etc.), the procedures around initial acknowledgement of receipt, and procedures around status updates until resolution of vulnerability.

5.3 – KEEP SOFTWARE UPDATED

Provision 5.3-2

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

Minimum Requirements

- The device shall have update mechanisms with anti-rollback policies to prevent users from being able to downgrade firmware (e.g., install an older version of the device's firmware, etc.).

- The device shall have update mechanisms that prevent users from using modified versions of firmware.

Supporting Evidence

The developer shall list all the device's update mechanisms. Examples of these update mechanisms are, but not limited to the following:

  - Automatic updates, where the device downloads and installs updates from a remote server,
  - Manual updates, where the user is required to download updates on the developer's website and manually install them on the device,
  - Computer tool provided by developer allowing the user to use a USB stick as the update medium,
  - Mobile application sends a request to download and install update from remote server,
  - Internet Service Provider (ISP) utilises fibre broadband to push firmware updates to the device, etc.

For devices that have manual update mechanisms, the developer shall provide evidence (e.g., screenshots, videos, etc.) to show the device can successfully reject the installation of both, older and modified versions of firmware.

Assessment

The assessor shall check if the developer has declared all the update mechanisms on the device (e.g., check if the device has a companion application that facilitates updates, etc.).

If the device has manual update mechanisms, the assessor shall check if the device is able to successfully reject the user's request to install both older and/or modified version of firmware.

Provision 5.3-3

An update shall be simple for the user to apply.

Minimum Requirements

- There shall be instructions provided to users to explain how they can apply updates to their devices using the device's update mechanisms.

Supporting Evidence

The developer shall provide evidence (e.g., user manuals, videos, etc.) to show that users are provided with instructions explaining how to apply updates to the device. Acceptable examples are, but not limited to:

  - How to toggle automatic updates,
  - How to apply update within the GUI and/or mobile application,
  - How to download firmware from developer's website and install it on the device.

Assessment

The assessor shall check if users are provided with instructions on how to apply updates to the device.

Provision 5.3-7

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

Minimum Requirements

- Data transmitted between the device and the update server (i.e., cloud) shall be encrypted with accepted cryptographic algorithms (refer to NIST SP 800-131A).

- Alternatively, the update transmission channel between the device and the update server shall be protected using TLS 1.2 or higher with acceptable cipher suites (refer to NIST SP 800-52).

Recommendation

For any applications where firmware checksums are applicable, developer should adopt the use of SHA-2 and any other accepted cryptographic algorithms mentioned in NIST SP 800-52.

Supporting Evidence

The developer shall provide evidence (e.g., Wireshark screenshot, tcpdump, etc.) showing that the data transmitted between the device and update server is encrypted, and clearly indicating both the source and destination IP addresses.

For constrained devices that do not support or require software/firmware updates, this provision may not be applicable. For such cases, the developer shall provide justification on why the device would not require support and/or software/firmware updates.

The developer shall also provide screenshots (e.g., testssl.sh script, cipherscan, etc.) to show that the transmission channel is using acceptable cipher suites in accordance with NIST SP 800-52.

Assessment

The assessor shall check the evidence provided to determine if the data transmitted is encrypted with acceptable cryptographic algorithms.

The assessor shall examine the evidence provided to ensure that the transmission channel between the device and update server is protected using TLS 1.2 or higher, with acceptable cipher suites (refer to NIST SP 800-52).

For constrained devices that do not require support and/or updates, the assessor shall check if the justification provided by the developer sufficiently explains why the device does not require support and/or updates.

Provision 5.3-8

Secure updates shall be timely.

Minimum Requirements

- During the device's specified support period, there shall be internal policies established by the developer that:
    - Considers the severity and criticality of security vulnerabilities and,
    - Specifies the expected period to provide security updates for each classification of vulnerabilities (i.e., critical, non-critical, high, medium, low, etc.).

Supporting Evidence

The developer shall declare the expected period for them to provide security updates based on the severity and criticality of the security vulnerability. An example of an acceptable declaration is provided below (this may vary based on the developer's internal vulnerability classification processes):

- Critical within X days/hours
- High within X days
- Medium within X days
- Low within X days

Where possible, the developer shall provide evidence (e.g., screenshots, documents, webpages, etc.) showing expected periods for providing security updates to address security vulnerabilities.

Assessment

The assessor shall check if the developer has internal policies that consider the severity of security vulnerabilities and has provided specified expected periods for resolving security vulnerabilities for the different vulnerability severities.

Provision 5.3-10

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

Minimum Requirements

- The device shall verify the authenticity and integrity of each update via a trust relationship (Refer to NIST SP 800-131A for all accepted cryptographic algorithms). The following examples are cases of establishing trust relationships, but are not limited to:
    - Firmware updates signed using digital signatures with an accepted cryptographic algorithm (e.g., RSA-4096-SHA512, etc.).
    - Key-pair authentication between the device and update server using accepted cryptographic algorithms.
    - Using SSL pinning.

- Specifically for manual updates, the trust relation must be established by using digital signatures with an accepted cryptographic algorithm (e.g., RSA-4096-SHA512, etc.).

(For devices that obtain updates that are pushed from a hub, the hub shall be responsible for verifying the authenticity and integrity of the firmware update.)

Supporting Evidence

The developer shall declare the kind of trust relationship established by the device to verify the authenticity and integrity of updates.

The developer shall declare the cryptographic algorithm used in the verification process (including instances where updates are pushed to devices from a hub).

Assessment

The assessor shall check that the device establishes a trust relationship that verifies the authenticity and integrity of updates.

The assessor shall check if the cryptographic algorithm declared by the developer is not deprecated and conforms to NIST SP 800-131A.

Provision 5.3-13

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

Minimum Requirements

- The device's support period shall be provided clearly on the developer's website.

- For subscriber-only ISP issued devices:
  The device shall be supported by the ISP (e.g., the device should still be able to receive firmware updates.) if user is no longer subscribed to the device issuing ISP's services,

  The device issuing ISP shall clearly indicate the support period of the issued device on their websites along with any relevant terms of use (e.g., procedures to comply with when user make changes to their subscriptions).

  If the device issuing ISP does not intend to provide support for nonsubscribers, the ISP shall have procedures in place to:
  - Have procedures in place to retrieve the device, or
  - Have measures in place to ensure that the device is not usable with another ISP.

Recommendation

To maximise the CLS(IoT) label's validity period of up to 3 years, developers can consider altering the device's support period shown on their websites as the CLS(IoT) label issued to them is valid only till the end of that specified support period.

An example would be if the developer's website states that the support period is for the next 2 years, the CLS(IoT) label issued for the device will only have a validity period of 2 years.

Supporting Evidence

The developer shall provide evidence (e.g., screenshots, support page URL, etc.) to show the device's defined support period stated on their website.

For subscriber-only ISP issued devices, the ISP shall provide evidence (e.g., screenshots, URLs, etc.) to show the device's defined support period stated on the ISP's website.

For instances where the device issuing ISP does not intend to continue supporting the device for non-subscribers, the ISP shall provide evidence (e.g., screenshots, URLs, guidance documents, etc.) of their device retrieval procedure or measures to ensure that the device is not usable with other ISPs.

Assessment

The assessor shall check if the device and its defined support period are published clearly on the developer's website.

For subscriber-only ISP issued devices, The assessor shall check the device's defined support period stated on the ISP's website.

For instances where the device issuing ISP does not intend to continue supporting the device for non-subscribers, the assessor shall check if the ISP has a device retrieval procedure in place or measures that ensure that the device cannot be used with other ISPs.

Provision 5.3-16

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

Minimum Requirements

- The product label shall clearly show the model designations (e.g., brand, product name, model number, product number, etc.) of the device.

- Devices shall have a unique identifier that allow users to differentiate and identify the device from similar ones. Some examples would include, but are not limited to:
    - smart switches with different gangs
    - routers with different colour
    - IP camera with different resolution / viewing angles

Supporting Evidence

The developer shall provide evidence in the form of screenshots, pictures or images of product labels that clearly show the product's model designation.

Assessment

The assessor shall examine the product labels provided to determine if they clearly show the device's model designations.

5.4 – SECURELY STORE SENSITIVE SECURITY PARAMETERS

Provision 5.4-1

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

Minimum Requirements

- Sensitive security parameters (e.g., passwords, tokens, secret keys, etc.) stored in persistent storage, shall be encrypted with a cryptographic algorithm in accordance with NIST SP 800-131A or via an appropriate secure storage mechanism.

Supporting Evidence

The developer shall declare if the following sensitive parameters are stored by the device:

- Administrative password,
- Device's private and public key used for client authentication,
- Root Key used to encrypt other sensitive parameters.

The developer shall list down any other sensitive security parameters (e.g., passwords, secret keys, etc.) stored in persistent storage that can impact the security of the device if exposed/compromised.

The developer shall describe the security mechanism(s) employed to store the sensitive security parameters declared earlier. Some examples of acceptable security mechanism descriptions are, but not limited to:

- If sensitive parameters are encrypted with a key, a description of how the encryption key is stored/derived is required,
- The sensitive parameters are stored in a secure element or tamper protected microcontroller rather than in external flash storage,
- A Trusted Execution Environment (TEE) is used to store and access sensitive parameters.

Assessment

The assessor shall check if the declared sensitive security parameters are protected (i.e., stored using secured storage mechanisms, encrypted, etc.).

Provision 5.4-2

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.

Minimum Requirements

- Any hard-coded unique per device identifiers used in the device for security purposes are implemented in a way that resists tampering.

Supporting Evidence

The developer shall declare if there are any hard-coded unique per device identifiers used in the device.

The developer shall describe how such declared hard-coded unique per device identifiers are implemented to resist tampering. Some examples of this are:

- Store hard-coded unique per device identity in the ROM.
- Burn hard-coded unique per device identity into one-time programmable hardware (e.g., eFuse, etc.).

The developer shall provide supporting evidence (e.g., pictures, screenshots, diagrams, etc.) where applicable.

Assessment

If there are any declared hard-coded unique per device identifiers used in the device, the assessor shall check that every hard-coded identity implemented on the device has an adequate security mechanism to resist tampering.

Provision 5.4-3

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

Minimum Requirements

- The device shall not contain hard-coded critical security parameters in the software source code. If there are hard-coded critical security parameters embedded within the device software source code, the device's provisioning mechanism shall ensure that such hard-coded critical security parameters are no longer in use during its operational phase.

Supporting Evidence

The developer shall declare if there are any hard-coded critical security parameters used in the device source code. If there are hard-coded critical security parameters embedded within the device software source code, the developer shall provide evidence to show how such hard-coded critical security parameters are no longer in use during the device's operational phase.

Assessment

The assessor shall check if the developer has declared that any hard-coded critical security parameters are within the device source code. If there are any declared, the assessor shall examine the evidence provided to ensure that hard-coded critical security parameters are no longer in use during the device's operational phase.

Provision 5.4-4

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.

Minimum Requirements

- Critical security parameters (e.g., secret keys, private components of certificates, etc.) used for integrity and authenticity checks of software/firmware updates shall be unique per device.

(There is no requirement for developer's public key to be unique per device.)

Supporting Evidence

The developer shall provide evidence to show that the critical security parameters used are unique per device.

The developer shall describe the generation process of the critical security parameters used by the device.

The developer shall provide evidence to show that the critical security parameters are produced with a mechanism that reduces the risk of automated attacks against classes of devices.

Assessment

The assessor shall examine the generation process of the critical security parameters, ensuring that they are unique per device and has a mechanism that reduces the risk of automated attacks against classes of devices.

5.5 – COMMUNICATE SECURELY

Provision 5.5-1

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

Minimum Requirements

- The communication (transmission channel) between the device and other services shall be established over TLS 1.2 or higher, with acceptable cipher suites (refer to NIST SP 800-52).

- For Wi-Fi routers, WPA2 or higher communication protocol shall be implemented while conforming to the best cryptographic practices for encryption algorithm as per NIST SP 800-131A.

- For Bluetooth communication (including BLE), it shall be configured as Security Mode 1 with Security Level 3 minimally (excluding Security Mode 2 with Security level 1).

- The IoT protocols used by the device for communication shall not be outdated and/or vulnerable (e.g., PnP, MQTT, etc.)

Supporting Evidence

The developer shall list down all communication channels between devices and other associated services. Examples of such communications are, but not limited to:

- Device to mobile app (companion app),
- Device to update server,
- Device to PC,
- Bluetooth connections,
- Usage of the ZigBee communication protocol.

The developer shall also provide evidence (e.g., screenshots, videos, documentations, whitepapers, etc.) to show that the device's communication channels are secure.

> For Wi-Fi routers, the developer shall provide evidence to show that communication protocol WPA2 or higher is enabled by default on the device (e.g., screenshot or video of the option in the GUI, tool scanner, etc.)

Assessment

The assessor shall check that the communication channels declared are accurate.

The assessor shall examine the evidence provided by the developer to ensure it is appropriate for secure communication.

Where applicable, the assessor shall check whether the cryptographic algorithm used are in accordance with NIST SP 800-131A.

Provision 5.5-5

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.

Minimum Requirements

- Users must be authenticated before any of the device's security relevant information are allowed to be changed. This includes authentication through mobile application, GUI, and even the CLI.

(Network service protocols [e.g., ARP, DHCP, DNS, ICMP, NTP, etc.] that are designed to enable external configuration without authentication are an exception for this provision.)

Supporting Evidence

The developer shall also provide the NMAP scan to show all open ports and associated services on the device. The command below can be used:

- *Port scan can be performed using nmap -sT -sU -A -p- <ip address>*

The developer shall list all authentication mechanisms (e.g., Telnet, FTP, SSH, device administrator portal, mobile application, Bluetooth, etc.).

The developer shall list all avenues where a user is able to make security-relevant changes and provide evidence (e.g., screenshots, videos of authentication prompts, etc.) to prove that security-relevant changes are only accessible after successful authentication.

Assessment

The assessor shall check if the list of authentication mechanisms the developer has provided is complete.

The assessor shall examine the NMAP scan to determine if the developer's list of authentication mechanisms is accurate.

Based on the list of avenues provided by the developer, the assessor shall check if security-relevant changes are only accessible after successful authentication.

Provision 5.5-7

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

(This provision is not applicable for cases where the device does not have remote access functionality.)

Minimum Requirements

- For devices that have remote access features, the communication (transmission) shall be conducted via TLS 1.2 or higher, with acceptable cipher suites (refer to NIST SP 800-52).

- For Wi-Fi routers, remote access features of the device shall be disabled by default (i.e., when a user first powers on the device, this function shall be disabled).
    - If remote access features (e.g., TR-069, etc.) is required for the Wi-Fi router to operate/function, the developer must obtain a waiver from IMDA.
    - When they are enabled, their communication must be over TLS 1.2 or higher.

Recommendation

It is recommended for remote access features to be disabled by default even if it is not a requirement under the CLS(IoT) (except for Wi-Fi routers).

Supporting Evidence

For devices that have remote access features, the developer shall provide evidence (e.g., Wireshark screenshot, etc.) to show that the communication is via TLS 1.2 or higher with acceptable cipher suites (refer to NIST SP 800-52).

For Wi-Fi routers, the developer shall declare if remote access (e.g., TR-069, etc.) to the device is disabled by default. If remote access features must be enabled for its operation, developers shall submit the waiver obtained from IMDA.

Additionally, for Wi-Fi routers, the developer shall provide evidence (e.g., Wireshark screenshots, etc.) to show that the remote access features of the device communicate via TLS 1.2 or higher with acceptable cipher suits (refer to NIST SP 800-52).

Assessment

The assessor shall check if remote access communication is minimally conducted over TLS 1.2 and with acceptable cipher suites.

For Wi-Fi routers, the assessor shall check if the remote access features are disabled by default. If they must be enabled for device operation, the assessor shall check if the developer has obtained a waiver from IMDA.

Additionally, for Wi-Fi routers, the assessor shall check that remote access features of the device communicate via TLS 1.2 or higher with acceptable cipher

suites.

Provision 5.5-8

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

Minimum Requirements

- There shall be secure key management processes for **all keys** related to the device functions stated below:
  - Secure communications
  - Secure firmware updates
  - Encryption of sensitive data and security parameters

- The secure key management processes shall encompass the following:
  - Key generation
  - Key provisioning
  - Key usage
  - Key storage
  - Key revocation
  - Key destruction

Recommendations

A lifecycle diagram/table can be used to provide better visual representation of the secure key management process that is implemented.

Supporting Evidence

The developer shall provide evidence (e.g., lifecycle diagram/table, etc.) to show that are secure key management processes for **all keys** related to the device functions stated above. An elaboration for each secure key management process can be found below:

- For key generation, the developer shall describe the process of key-pair generation and state the cryptographic key generation library or tool used (e.g., HSM).

- For key provision, the developer shall explain how the private keys are provisioned to the firmware signing server or used during the firmware signing process.

- For key usage, the developer shall describe how the keys are used and how access to the key is controlled.

- For key storage, the developer shall elaborate on the security storage of private keys(e.g., using HSM, Android keystore, etc.).

- For key revocation, the developer shall describe the process when a key should be removed from operational use prior to the end of the established cryptoperiod of that key.

- For key destruction, the developer shall provide details of how the key is securely erased such that it cannot be recovered through physical or electronic means when the key is no longer used.

Assessment

The assessor shall examine the evidence to determine if the developer's description of the secure key management process is comprehensive, considering its generation, provisioning, storage, usage, and destruction.

5.6 – MINIMISE EXPOSED ATTACK SURFACES

Provision 5.6-1

All unused network and logical interfaces shall be disabled.

Minimum Requirements

- The device shall disable all unused and unnecessary (not essential to device operation) network and logical interfaces.

Supporting Evidence

The developer shall provide a list of all network and logical interfaces of the device (i.e., Bluetooth, ZigBee). Additionally, the developer shall declare if these interfaces are used or unused.

The developer shall provide evidence (e.g., Ubertooth scan results, configuration files, etc.) to show that all unused and disabled interfaces are indeed disabled.

The developer shall perform a port scan (refer to nmap command below) of the device for both WAN and LAN, then provide evidence (e.g., screenshots, attaching the nmap output file, etc.) showing the full report, ensuring that all ports are accounted for.

- *Port scan can be done using nmap -sT -sU -A -p- <ip address>*

The developer shall provide justification for all open/enabled ports. An example of this is shown below:

- Open Port: Justification
- Port 8443: HTTPS client authentication connection

Assessment

The assessor shall check if the list of network and logical interfaces provided by the developer is complete.

The assessor shall examine the output of the port scan to determine if there are any unaccounted ports or open ports that are unused.

Provision 5.6-2

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

Minimum Requirements

- The device shall minimise the unauthenticated disclosure of security-relevant information that could aid a threat actor in compromising the device.

- If there are any unauthenticated disclosure of security-relevant information while the device is in the initialised state, it shall be deemed to be necessary for device functionality and/or needed for authentication.

Supporting Evidence

The developer shall declare all security-relevant information that will be disclosed while the device is in the initialised state (unauthenticated) and explain how they are necessary for device functionality and/or needed for authentication.

The developer shall provide supporting evidence (e.g., submission of nmap report, screenshot of nmap report, etc.) to account for all disclosed security-relevant information by performing the following command:

- *nmap -sT -sU -A -p- <ip address>*

Assessment

The assessor shall check if the explanation provided by the developer for the disclosure of all security-relevant while the device is in the initialised state is appropriate.

The assessor shall examine the evidence provided to account for all disclosed security-information and determine if there are potentially any other instances of unauthenticated disclosure of security-relevant information on the device.

Provision 5.6-4

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

Minimum Requirements

- All debug interfaces shall be disabled; this can only be achieved by disabling in software.

Supporting Evidence

The developer shall provide evidence (e.g., screenshot of configuration, unsuccessful connection attempt, etc.) showing that the debug interfaces are disabled on the device.

Assessment

The assessor shall check if debug interfaces are confirmed to be disabled on the device.

5.8 – ENSURE THAT PERSONAL DATA IS PROTECTED

Provision 5.8-2

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

Minimum Requirements

- Communication of sensitive user data between device and associated services shall be conducted over TLS 1.2 or higher, with acceptable cipher suites (refer to NIST SP 800-52).

Supporting Evidence

The developer shall provide a list of all avenues where sensitive user data could be communicated (transmitted) and evidence (e.g., Wireshark screenshot, etc.) to show that this communication is secure. Where possible, the developer should specify the source and destination IP addresses.

Additionally, the developer shall also declare the cryptographic algorithm used to encrypt the data collected and/or transmitted.

Assessment

The assessor shall check if the communication (transmission channel) used to transfer the data collected by the device is secure.

The assessor shall also check if the cryptographic algorithm used to encrypt the sensitive user data is in accordance with NIST SP 800-131A.

Provision 5.8-3

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

Minimum Requirements

- All the device's external sensing capabilities shall be well documented and shall be easily accessible to the user (i.e., via prompts, documentations, alerts, etc.). This also includes external sensing capabilities that may be disabled by default (e.g., microphone, Bluetooth, etc.).

(External sensing capabilities of devices includes the use of optical, acoustic, biometric, or location sensors).

Supporting Evidence

The developer shall declare if the device has any external sensing capabilities, including external sensing capabilities that may be disabled by default.

The developer shall provide evidence (e.g., screenshots, attachments, user guidance documents, etc.) to show how the device's external sensing capabilities are communicated to the user.

Examples of a device with external sensing capabilities are, but not limited to,

- IP camera device with external sensing capabilities such as an audio microphone.
- Smart watch device with external sensing capabilities such as a GPS.

Assessment

The assessor shall check if the developer has declared if the device has any external sensing capabilities.

If so, the assessor shall check if the information regarding the device's external sensing capabilities is provided to the user.

5.11 – MAKE IT EASY FOR CONSUMERS TO DELETE PERSONAL DATA

Provision 5.11-1

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

Minimum Requirements

- The device shall have at least one feature (e.g., through GUI, companion mobile application, hardware reset button, etc.) that allows the user to erase personal data, user configuration settings, and cryptographic material (e.g., user passwords, keys, etc.) that are stored on it.

- Information about th feature(s) shall be made known to explain to users how to perform actions required to delete data stored on the device.

Supporting Evidence

The developer shall list all the features that erases user data stored on the device.

The developer shall provide evidence (e.g., user guidance documents, screenshots of webpages that show users how to erase personal data on device, etc.) to show the feature exists and is made known to the user.

Assessment

The assessor shall check if the device has at least one feature that erases user data from the device.

The assessor shall examine the evidence to determine if these features are made known to the user.

5.13 – VALIDATE INPUT DATA

Provision 5.13-1

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

Minimum Requirements

- Input data shall be validated for all interfaces of the device (including companion applications and other associated services), to reduce unexpected behaviour from occurring during data processing.

Recommendations

For specifics and best practices related to input data validation, please refer to the Input Validation Cheat Sheet by OWASP.

Additionally, to reduce the risk of overexposing data and to prevent access to core functionalities, the device interface should be limited to only those required for the device's operation.

Supporting Evidence

The developer shall provide a list of all the interfaces of the device that accepts input data and processes them (e.g., the device administrator portal, the login page of the companion application, etc.).

The developer shall also provide evidence (e.g., screenshots, videos, etc.) to show that the device has input data validation by rejecting invalid and malformed requests. Alternatively, the developer shall describe the input validation mechanism(s) implemented on all device interfaces declared earlier.

Assessment

The assessor shall check if the list of interfaces provided by the developer is complete (i.e., verify if all login interfaces within the device, its companion application and associated devices are accounted for).

The assessor shall examine the evidence to determine if the device is able to reject invalid and malformed requests made. This shall be verified minimally for the interfaces of the device where authenticated, allows the change of authentication values and/or information.

6.1 – DATA PROTECTION PROVISIONS FOR CONSUMER

Provision 6.1

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.

(Please note that the information provided by the developer in this clause will be provided alongside the product's listing on the CSA website.)

Minimum Requirements

- The developer shall provide users with information about what personal data is processed, how it is being used, by whom, and for what purposes, for the device, its companion application, and other associated services.

(These requirements also apply to third parties that can be involved such as advertisers, partnered service providers, etc.)

Supporting Evidence

The developer shall list all personal data collected by the device, its companion application, and other associated services. Where possible, describe how the collected data is encrypted/protected when stored or transferred by the device.

The developer shall provide evidence (e.g., screenshots, videos, documents, emails, etc.) to show that users are provided with information about what personal data is processed, how it is being used, by whom, and for what purposes. The evidence provided here should cover the following scenarios:

- Processing of personal data,
- Collection of personal data,
- Processing of telemetry data.

Possible examples of acceptable evidence are, but not limited to:

- Pop-up notifications when the user enables a feature on the device that clarifies the data that is going to be collected upon activation,
- Terms of use webpage or document with relevant sections highlighted, Privacy policy webpage or document with relevant sections highlighted.

Assessment

The assessor shall examine the evidence and determine if it shows that users are provided with information about what personal data is processed, how it is being used, by whom, and for what purposes.

Provision 6.2

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

Minimum Requirements

- For all cases where a user's personal data is going to be processed, the user's consent shall be obtained in a manner where they are notified on what personal data is being processed and for what purpose (i.e., a GPS feature requiring the user's location via the device's location services, etc.).

Supporting Evidence

The developer shall provide evidence (e.g., screenshot of prompts, pop-up alerts, privacy policies, etc.) to show how a user's consent is obtained before processing their personal data. The evidence shall clearly indicate what personal data is being processed.

Assessment

The assessor shall check if the evidence provided by the developer shows what personal data is being requested for when the device requests consent from the user.

Examples of acceptable evidence are, but not limited to:

- Pop-up when user enables a feature on DUT to inform them what data is going to be collected upon activation,
- Terms of use document or webpage,
- Privacy policy document or webpage.

Provision 6.3

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

Minimum Requirements

- The device, its companion application, and other associated services (where applicable) shall allow the user to withdraw from the collection and processing of their personal data at any time, even if they have given consent to it previously.

Supporting Evidence

The developer shall provide evidence (e.g., screenshots of withdrawal function in GUI, submitting a request through the product webpage to request for a withdrawal, guidance documents to explain how to withdraw user's information from being processed, etc.) to show that users can withdraw from the collection and processing of their personal data at any time.

Assessment

The assessor shall examine the evidence to determine if users are able to withdraw from the collection and process of their personal data at any time, even if they have given consent to it previously.

Provision 6.5

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.

Minimum Requirements

- If the device collects telemetry data, there shall be information provided to inform the user on what telemetry data is collected, how it is being used, by whom, and for what purposes.

Examples of telemetry data but are not limited to:

- Authentication and authorisation,
- System processes,
- System file changes,
- User behaviour.

Supporting Evidence

The developer shall declare if the device, its companion application, and associated services (e.g., Mobile application, GUI, peripherals, etc) collect telemetry data.

If the device collects telemetry data, the developer shall explain what is collected, how the data is being used, by whom, and for what purposes. The developer shall also provide evidence (e.g., screenshots, videos, documents, etc.) to show how this information is provided to users.

Assessment

If the device, its companion application, and associated services collect telemetry data, the assessor shall check if the user is provided with information on what telemetry data is collected, how it is being used, by whom and for what purpose.

LIFECYCLE DOCUMENTS

CK-LP-01

Have you conducted threat modelling to identify, analyse and mitigate threats to the device?

Minimum Requirements

- Threat modelling shall be performed to identify, analyse, and mitigate threats to the device.

Supporting Evidence

The developer shall provide evidence (e.g., internal software development lifecycle process documents, process diagrams, etc.) to show that the threat modelling process is part of the device's development lifecycle. The threat modelling process shall include the following:

- Defining the security problem.
- Conduct risk assessment.
- Determine the security objectives.
- Define the security requirements.
- Design and implement.
- Validate and verify that the device security capabilities address the security requirements.

Alternatively, the developer may also provide the results of the threat modelling process performed on the device as evidence. For this case the evidence shall include the security problem, the security objectives (desired outcomes) and subsequently describe how the implemented security capabilities help to mitigate risks identified during the threat modelling process.

Assessment

The assessor shall examine the threat modelling performed by the developer and verify if it includes the key points stated in the minimum requirements.

The assessor shall also check if the security features declared by the developer is sufficient in mitigating risks raised from the threat modelling process.

CK-LP-02

Did you design and develop the device using a secure engineering approach?

Minimum Requirements

There shall be at **least one** secure engineering approach that falls within each of the two categories, namely, Development Practices and Testing (refer below for examples) adopted in the development of the device, including its companion application and other associated services, where applicable.

Development Practices

No.	Secure Engineering Approach (Non-exhaustive):	Examples, but are not limited to:
1	Secure coding practices	• Validation of all inputs, outputs with proper encoding • Detecting and handling errors • Avoiding the use of unsafe functions and calls • Secure coding standards and guidelines • Source code obfuscation
2	Reuse of existing well-secured software libraries	• How deployed software components are updated • Use of code repository to store and maintain secure software for reuse when suitable
3	Using non-default, well-organised compiler, interpreter and build process	• Use of compilers, interpreters and build tools that offer features such as the randomisation of memory location usage to improve executable security
4	Others	• Entered characters for passwords are masked with special characters and not displayed as plaintext • Locking or termination of inactive session after a certain period, and prevention of reusing session information upon reconnection.

Testing

No.	Secure Engineering Approach (Non-exhaustive):	Examples, but are not limited to:
1	Functional testing on security features	• Static Application Security Testing (SAST) • Dynamic Application Security Testing (DAST)

		• API testing • Fuzz testing
2	Developer and/or peer code review system	• The use of tracking system for code review, feedback and provide remediation status of the findings raised
3	Software deployed in secure settings by default	• Document reflecting each device setting's purpose, options, default values, security relevance, operational impact, and relationships with other settings
4	Others	

Supporting Evidence
The developer shall declare what secure engineering approach(es) were adopted in the development of the device (including its companion application, and other associated services, if applicable).
To show that secure engineering approaches were adopted in the development of the device, the developer shall provide evidence in the form of, but not limited to the following:

- Process/guidance documents that shows the steps/processes performed internally by the developers to increase device security.
- Screenshots of tools that the developers used for internal tracking for findings raiders and code review (e.g., JIRA, etc.).
- Provide penetration test, static/dynamic analysis reports, etc. Or screenshot of relevant portions of these reports.

Assessment
The assessor shall check if at **least one** secure engineering approach from both the Development Practices and Testing categories were adopted in the development of the device (including its companion application, and other associated services, if applicable).

The assessor shall check if proper testing methods and tools were used to determine the applicability of the tests performed.

CK-LP-03

Do you implement and maintain the device with components from a secure supply chain, with no known unmitigated vulnerabilities?

Minimum Requirements

- Proper measures shall be in place to ensure that the device components (for both software and hardware components) have no known vulnerabilities that are unmitigated.
- There shall also be practices in place to ensure patching of outdated libraries.
- Vulnerable libraries (including libraries from third-party sources) shall not be used in the released version of the device.
- There shall be practices in place that evaluate and select device components to be used in the production version of the device.
- If there are security vulnerabilities in the operating system, support shall be provided so that these security vulnerabilities are addressed through manual or automatic updates.

Supporting Evidence

The developer shall provide the device components list (e.g., Software Bill of Materials (SBOM), Hardware Bill of Materials (HBOM), etc.). Non-security related components can be omitted for HBOM submissions.

The developer shall describe (e.g., assessment criteria, performance monitoring, etc.) how these device components are evaluated and selected to be used in the production version of the device.

The developer shall provide evidence (e.g., internal process documents, whitepapers, etc.) to show the processes for patching vulnerable and outdated libraries (including third-party libraries).

The developer shall provide evidence (e.g., internal process documents, whitepapers, etc.) to show the processes for patching security vulnerabilities in the operating system.

Assessment

The assessor shall check if the developer provided a device component list.

The assessor shall check if the developer has processes that evaluate and select the components used in the production of the device to reduce the chance of components having any known unmitigated vulnerabilities.

The assessor shall check if there are processes in place to ensure regular patching of vulnerable and outdated libraries (including third-party libraries) used by the device.

CK-LP-04

Do you provide, communicate, and update security information (terms of service, features, guidelines, instructions, and notifications, etc), in simple language and timely manner?

Minimum Requirements

- There shall be security information (e.g., information disseminated to users regarding updates, information provided to users about the risks mitigated by updates, etc.) provided to the users of the device.

Supporting Evidence

The developer shall provide evidence (e.g., screenshots, documentation, webpages, etc.) to show security information is communicated to users. Possible examples are, but not limited to:

- Official statements published on the developer website,
- Advisories posted on the developer's social media accounts,
- Notification on device interfaces.
- Device changelogs
- Patch notes

Assessment

The assessor shall check if security information is communicated to users.

CK-LP-05

Do you ensure that the device is hardened prior to release?

Minimum Requirements

- The device shall be hardened prior to release. Examples of device hardening measures are, but not limited to:
  - Closing of unused ports and services,
  - Removing all backdoors,
  - Removing all debug codes from released version,
  - Unnecessary accounts must be removed,
  - Unnecessary network interfaces must be disabled,
  - Enabling of security function of the operating system,
  - Blocking unauthorized network traffic or traffic transmitted from devices not registered in the network access control policy.

Supporting Evidence

The developer shall describe how device hardening was performed and provide documentation of instructions/processes that their development teams followed to perform them.

The developer shall perform a nmap scan and provide the output as evidence (e.g., screenshots, nmap output file, etc.) to show that all unused ports are closed.

  - *Port scan can be done using nmap -sT -sU -A -p- <ip address>*

If applicable, the developer shall provide evidence (e.g., screenshots, scan results, etc.) to show that scanning tools were used for checking debug/backdoor codes or hard-coded security credentials are removed in the device's production version.

If applicable, the developer shall provide evidence (e.g., screenshots, documentations, pictures, etc.) to show that all physical debugging ports (such as UART) are physically disabled and inaccessible. (We recommend using PuTTy to obtain this evidence).

Assessment

The assessor shall check if device hardening was performed according to the instructions/processes provided by the developer.

The assessor shall examine the nmap report and confirm that all unused ports are closed.

If applicable, the assessor shall examine the evidence to determine if debug/backdoor codes or hard-coded security credentials are removed in the device's production version.

If applicable, the assessor shall examine the evidence to determine if physical debugging ports (such as UART) are physically disabled and inaccessible.

CK-LP-06

Do you maintain an inventory of components including its version, applied patches and updates?

Minimum Requirements

- There shall be a well-maintained inventory of components which includes versions, applied patches and updates. Examples of accepted device component inventory are, but not limited to:
  - Software Bill of Materials (SBOM),
  - Hardware BOM,
  - Mobile application BOM,
  - Version control system.

Supporting Evidence

The developer shall provide the device's bill of materials, minimally indicating the components used, version of the components, and its license.

If applicable, developer shall provide the mobile application BOM and version control system (e.g., subversion, git, etc.).

Assessment

The assessor shall check if a well-maintained inventory of components is tracked and presented in a suitable format that minimally indicates the components used, along with their version numbers and corresponding licenses.

CK-LP-07

Do you conduct penetration testing and/or vulnerability assessment periodically, and before each major release?

Minimum Requirements

- There shall be penetration testing and/or vulnerability assessment conducted periodically, and before the release of major updates.

Supporting Evidence

The developer shall declare if penetration testing and/or vulnerability assessments are internally conducted periodically, and before the release of major updates.

The developer shall provide evidence (e.g., internal documents, VA/PT reports, etc.) to show how these tests are performed, by whom, how frequent, and the tools used to perform these tests.

Assessment

The assessor shall examine the evidence provided to confirm that these tests and assessments have been conducted on the device periodically and before major updates.

The assessor shall check if the tools listed by the developer can conduct valid penetration tests and/or vulnerability assessments.

CK-LP-08

Do you establish proper vulnerability disclosure and management?

Minimum Requirements

- Proper vulnerability classification shall be performed using the CVSS v3 (or higher) system for the classification of vulnerabilities into different severity ratings (critical, high, medium, low).

- The developer shall have sufficient supply chain capability to ensure that the components (for both hardware and software) they use in their device are updated when necessary.

Supporting Evidence

The developer shall describe the internal processes (e.g., how device vulnerabilities are reported, plans to ensure that these vulnerabilities are addressed) for managing the device's patches and updates when a vulnerability is reported to them.

The developer shall then provide evidence (e.g., screenshot of the CVSS rating of the vulnerability described earlier, etc.) to show that the CVSS v3 (or higher) is used in the classification of vulnerabilities into severity ratings.

The developer shall also provide internal documents that describes their supply chain capability, ensuring that components used in their device are continuously up to date with patches and updates provided.

Assessment

The assessor shall check if there are internal processes for the management of the device's patches and updates when a vulnerability gets reported.

The assessor shall check if CVSS v3 system (or higher) is used in the classification of device vulnerabilities.

The assessor shall check if the internal documents that describes the developer's supply chain capability are accurate.

REFERENCES

[1]	ETSI, “Cyber Security for Consumer Internet of Things,” ETSI EN 303 645.
[2]	ETSI, "Cyber Security for Consumer Internet of Things: Conformance Assessment of Baseline Requirements", ETSI TS 13 701.
[3]	ETSI, "Guide to Cyber Security for Consumer Internet of Things", ETSI TR 103 621.
[4]	Info-communications Media Development Authority of Singapore, “IMDA Internet of Things (IoT) Cyber Security Guide”.
[5]	NIST, "Guidelines for the Selection, Configuration, and use of Transport Layer Security (TLS) Implementations", NIST SP 800-52.
[6]	NIST, "Transitioning the Use of Cryptographic Algorithms and Key Lengths", NIST SP 800-131A.
[7]	Cyber Security Agency of Singapore, “CLS(IoT) Publication #2 - Scheme Specifications, CCC SP-151-2,” Version 1.3, September 2023.
[8]	Cyber Security Agency of Singapore, “CLS(IoT) Publication #3 - Requirements for Testing Laboratory,” Version 1.1, September 2023.

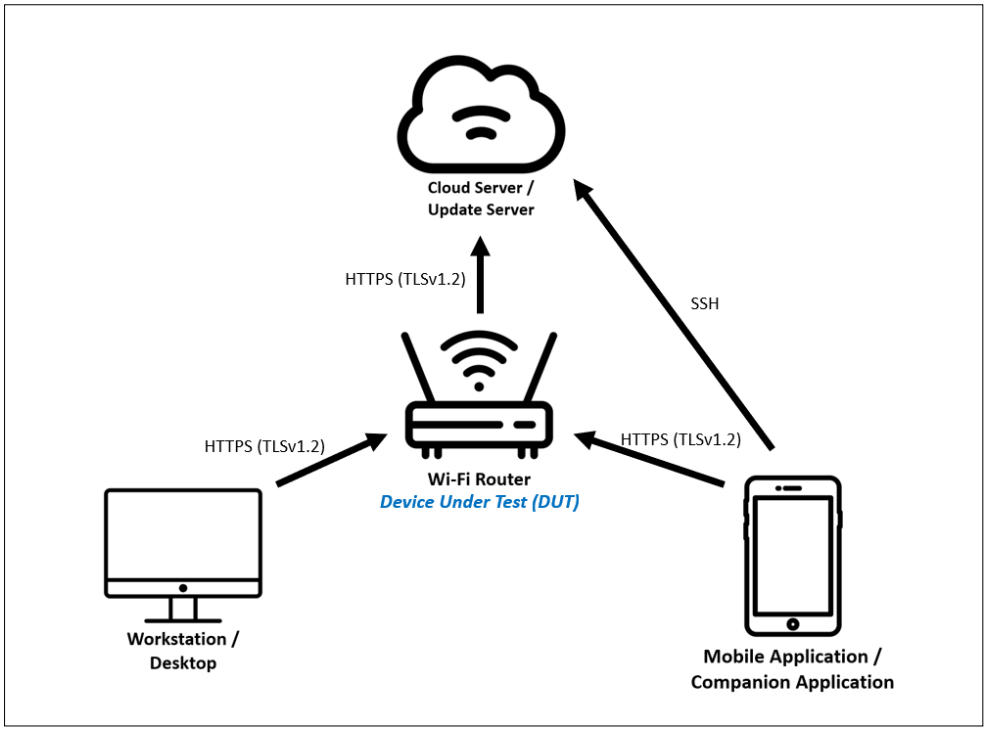
ACRONYMS

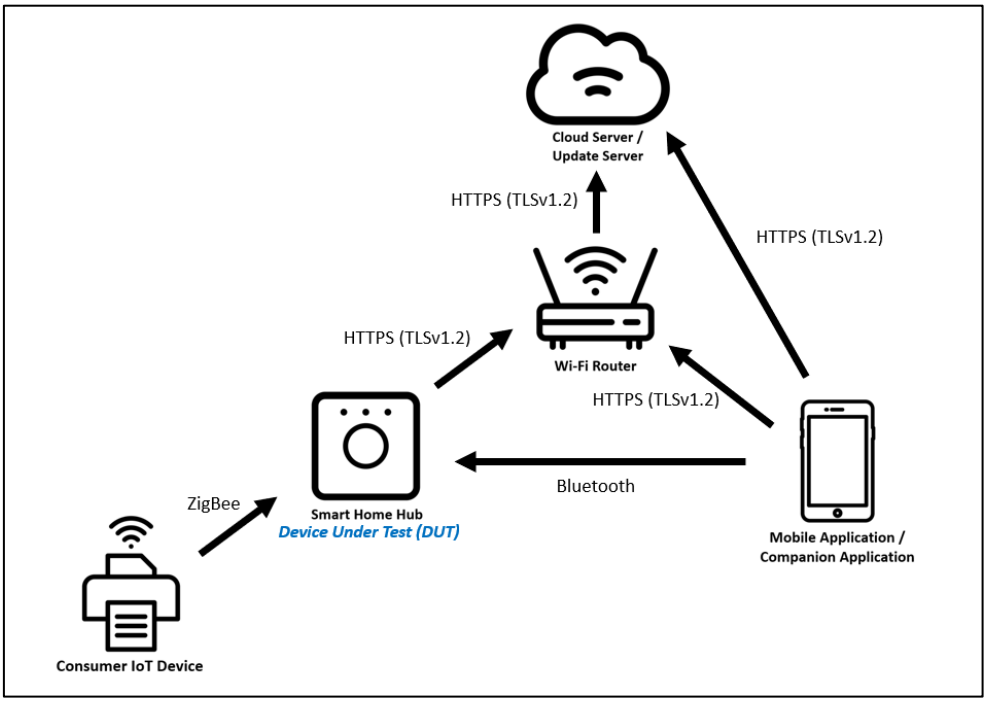
The following acronyms are used in CLS(IoT) Publication No. 4:

CCC	Cybersecurity Certification Centre
CLS	Cybersecurity Labelling Scheme
CSA	Cyber Security Agency of Singapore
DUT	Device Under Test
ETSI	European Telecommunications Standards Institute
GUI	Graphical User Interface
HPL	Historical Product List
IMDA	Info-communications Media Development Authority
IoT	Internet of Things
LPL	Labelled Product List
TL	Testing Laboratory
CLI	Command Line Interface
CVE	Common Vulnerabilities and Exposures
CVSS	Common Vulnerability Scoring System
API	Application Programming Interface

ANNEX A – NETWORK DIAGRAM

The purpose of Annex A is to provide clarification on the supporting evidence required for Security Provision 5.1-3. The following examples are acceptable examples of diagrams where the entities (e.g., device, cloud, hub, etc.) are highlighted and the sequence of authentication can be clearly seen.





ANNEX B – SUPPORTING EVIDENCE FOR TLS

In cases where TLS is used for secure communication. The manufacturer shall identify whether the device functions as the client or the server in the TLS connection.

If the device functions as the client, the manufacturer shall provide a Wireshark screenshot showing the following information:

1) Source and destination IP address
2) Open a "Client Hello" packet from this specific source and destination IP address to show the following (as indicated in reference image below):
    a. TLS version
    b. Cipher Suites

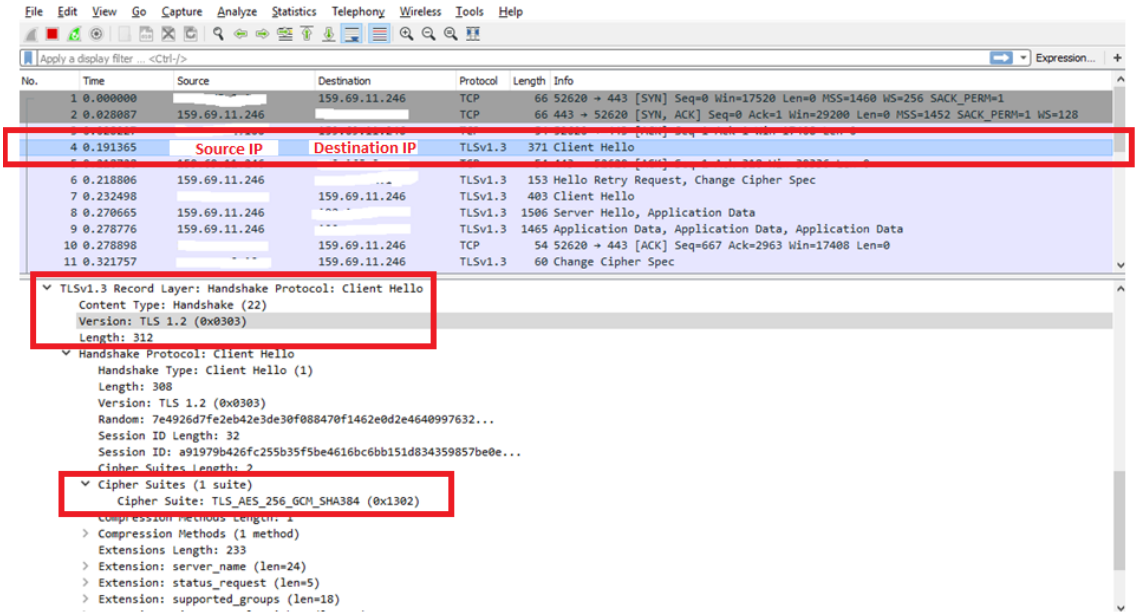


Image 1 – Highlighted information that should be included in the Wireshark screenshot

If the device functions as the server, the manufacturer shall provide a screenshot of the testssl.sh output showing supported cipher suites. Refer to image below.

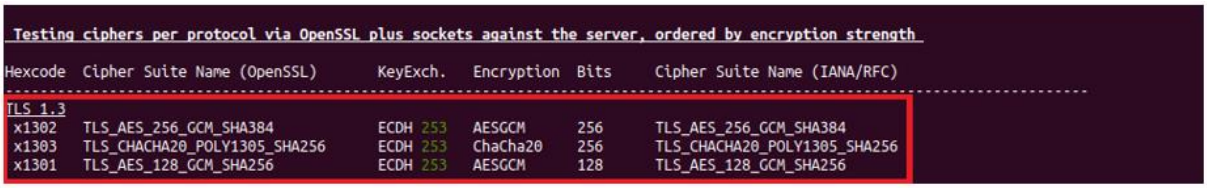


Image 2 – Highlighted information that should be included in the testssl.sh screenshot

For devices that use publicly accessible authentication servers (e.g., Identify Providers, Authentication APIs, etc.) for user authentication, Qualys SSL Server Test may be used to provide supporting evidence.

The manufacturer shall provide the following:

1) URL to the publicly accessible authentication server.
2) Use the Qualys SSL Server Test to generate a SSL report and save it in PDF format. This can be obtained by selecting to print the page from the browser and choosing to save it as PDF.