Cybersecurity Labelling Scheme

BY CYBER SECURITY AGENCY OF SINGAPORE

Cybersecurity Labelling Scheme (CLS)
Publication No. 2

Scheme Specifications

October 2020
Version 1.0

FOREWORD

The Cybersecurity Labelling Scheme (CLS) is part of Cyber Security Agency's (CSA) efforts to better secure Singapore's cyberspace and to raise cyber hygiene levels.

Under the CLS, the cybersecurity label would provide an indication of the level of security in the network-connected smart devices. It aims to improve security awareness by making such 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.

The CLS 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 with the objective of eliminating duplicated assessments across national boundaries.

The CLS 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 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	October 2020	Cyber Security Agency of Singapore	Release

CONTENTS

1 INTRODUCTION

1.0.1 This document aims to provide an overview of Cybersecurity Labelling Scheme (CLS) scheme. It outlines the four (4) tiers of assessment, the conformance checklist, testing activities, acceptance criteria, and the expected deliverables of each of the tiers.

1.0.2 The intended audience for this document is the developers who are interested in getting their Internet-Connected Devices labelled under CLS and testing laboratories who are responsible for testing the devices in accordance to the requirements of the CLS.

1.0.3 This document is organised in the following manner:

a. Chapter 2 provides a broad overview of the 4 tiers of assessment required under the different labelling levels of the CLS.

b. Chapter 3 elaborates on Assessment Tier 1 – Declaration of Conformance to Security Baseline Requirements. It lists the objective, requirements, and the acceptance criteria.

c. Chapter 4 elaborates on Assessment Tier 2 – Declaration of Conformance to Lifecycle Process Requirements. It lists the objective, requirements, and the acceptance criteria.

d. Chapter 5 elaborates on Assessment Tier 3 – Software Binary Analysis. It lists the requirements, test scope, pass criteria, and the test deliverables expected by CCC.

e. Chapter 6 elaborates on Assessment Tier 4 – Penetration Testing. It lists the requirements, test scope, pass criteria, and the test deliverables expected by CCC.

f. Chapter 7 contains the Conformance Checklist that is required for Assessment Tier 1 and 2.

1.0.4 The following roles are commonly referred in this document:

1. **Developer** of the Device Under Test (DUT)
2. **Testing Laboratory (TL)** that performs the Assessment Tier 3 and 4
3. **Cybersecurity Certification Centre (CCC)** that oversees the CLS

1.0.5 The CLS references the following documents:

1. The ETSI EN 303 645 – Cyber Security for Consumer Internet of Things [1] produced by the European Telecommunications Standards Institute (ETSI). The document outlines a set of outcome-focused security provisions to support developers in ensuring that their IoT products are secure by focusing on technical controls and organizational policies that matter most in addressing the most significant and widespread security

shortcomings.

2. The IMDA Internet of Things (IoT) Cyber Security Guide [2] produced by the Info-communications Media Development Authority of Singapore (IMDA). The document provides baseline recommendations, foundational concepts and checklists, which focus on the security aspects for the development, operations and maintenance of IoT.

2 OVERVIEW

2.1 CYBERSECURITY LABELING SCHEME (CLS)

2.1.1 The following table provides an overview of the broad requirements for each labelling level of the CLS.

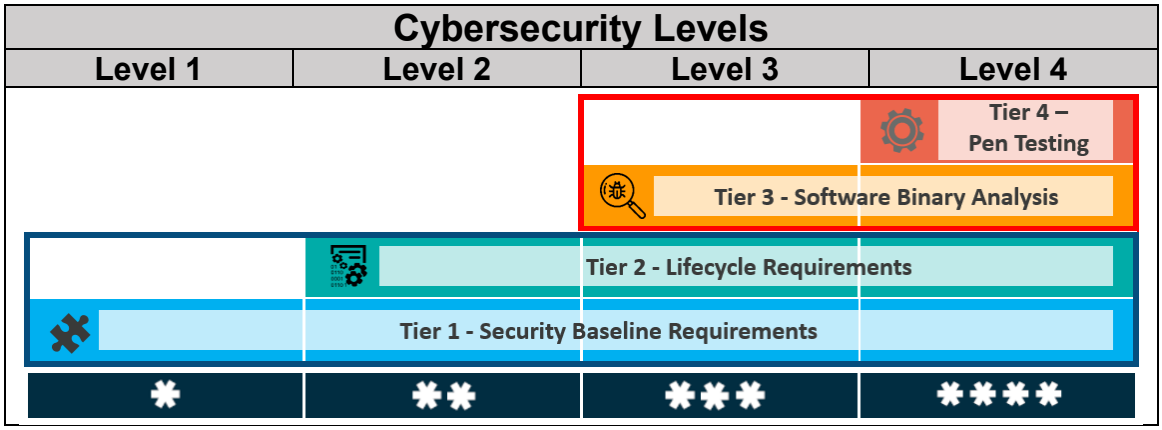

Table 1 - Cybersecurity Levels and Assessment Tiers

3 ASSESSMENT TIER #1 – SECURITY BASELINE REQUIREMENTS

3.1 OBJECTIVE

3.1.1 The objective of this assessment tier is to ensure that the Device Under Test (DUT) conforms to a minimal set of security baseline requirements.

3.1.2 **Assessment Tier #1 is based solely on declaration of conformance by the developer**.

3.1.3 Devices that have completed Assessment Tier 1 would entail that the developer has taken steps to mitigate against common basic attacks and IoT security problems, namely, avoiding the use of universal default password, by keeping device software updated, and by having a vulnerability disclosure policy to manage vulnerability reporting.

3.2 REQUIREMENTS

3.2.1 Assessment Tier #1 references the set of outcome-focused security categories specified within the ETSI EN 303 645 – Cyber Security for Consumer Internet of Things [1].

3.2.2 Depending on the level of the Cybersecurity Label that the developer wishes to attain, the number of provisions that are mandatory increases. Non-conformance to provisions categorised as “Mandatory” shall lead to the failure of this activity.

CLS Level	No. of Mandatory Provisions	Format
Level 1	13 of 67 provisions	Submission of Conformance Checklist and supporting evidences.
Level 2	21 of 67 provisions	Conformance Checklist and supporting evidences to be reviewed by CCC.
Level 3	24 of 67 provisions	
Level 4	32 of 67 provisions	

Table 2 - Assessment Tier #1 Requirements

3.3 DECLARATION OF CONFORMANCE

3.3.1 Developers are required to submit the Conformance Checklist found in Chapter 7 and required supporting evidences to CCC to declare conformance to the security requirements.

3.3.2 Some examples of supporting evidences include detailed descriptions, screenshots, process charts, work instructions. The expected supporting evidences are listed in the Conformance Checklist.

3.4 ACCEPTANCE CRITERIA

3.4.1 No independent testing by the testing laboratory is required for this assessment tier.

3.4.2 At CLS Level 1, the CLS label is awarded upon acceptance of the duly completed Conformance Checklist.

3.4.3 However, at CLS Level 2 and above, the Conformance Checklist and supporting evidences are reviewed by the CCC prior to approval, and Assessment Tier #1 is only considered satisfied when CCC gains assurance through the submitted supporting evidences that the requirements are met.

3.4.4 Where necessary, CCC may choose to request for further clarifications or a presentation from the developer.

3.4.5 In the event of non-conformities, the developer may choose to resolve them, or the application shall be considered as unsuccessful.

3.4.6 Should any false declarations be subsequently discovered (possibly by the TL in subsequent testing or by other means), CCC reserves the full rights to enforce actions as described in Chapter 8.7 of CLS Publication #1 – Overview of the Scheme [3].

4 ASSESSMENT TIER #2 – LIFECYCLE REQUIREMENTS

4.1 OBJECTIVE

4.1.1 The objective of this activity is to ensure that the developer adopts a "Security-by-Design" approach and implements adequate processes and practices to design, create, and maintain security in the Internet-Connected Device.

4.1.2 **Assessment Tier 2 is based solely on declaration of conformance by the developer**.

4.1.3 Devices that complete Assessment Tier 2 would entail that the developer has taken steps to identify the threats commonly associated with such devices and have implemented security measures against common threats for Tier 2.

4.2 REQUIREMENTS

4.2.1 Assessment Tier 2 references the lifecycle security considerations of the IMDA IoT Cyber Security Guide [2] published by the Info-communications Media Development Authority (IMDA).

4.2.2 The developer is required to fulfil all 9 lifecycle provisions (CK-LP-01 to CK-LP-09) listed in Chapter 7 – Conformance Checklist of this document.

4.3 DECLARATION OF CONFORMANCE

4.3.1 For **all** device categories, the developer shall complete and submit the Conformance Checklist found in Chapter 7 – Checklist of this document to CCC to declare conformance to lifecycle requirements.

4.3.2 The developer shall provide adequate supporting evidences alongside the Conformance Checklist (e.g. detailed descriptions, screenshots, process charts, work instructions, etc.) such that CCC is able to assess if the requirements for Tier 2 have been met, and that the security lifecycle processes and practices are adopted. Some examples of the expected supporting evidences are listed in the checklist.

4.4 ACCEPTANCE CRITERIA

4.4.1 No independent testing nor an audit by the testing laboratory is required for Assessment Tier #2.

4.4.2 However, CCC will review the submitted Conformance Checklist and supporting evidences. Assessment Tier #2 is only considered satisfied when CCC gains assurance through the submitted supporting evidences that the developer has implemented the required processes and practices and utilises them throughout the lifecycle of the DUT.

4.4.3 Where necessary, CCC may choose to request for further clarifications or a presentation from the developer.

4.4.4 In the event of non-conformities, the developer may choose to resolve them, or the application shall be considered as unsuccessful for Level 2.

4.4.5 Should any false declarations be subsequently discovered (possibly by the TL in subsequent testing or by other means), the testing laboratory are to inform the CCC, and CCC reserves the full rights to enforce actions as described in Chapter 8.7 of CLS Publication #1 – Overview of the Scheme [3].

5 ASSESSMENT TIER #3 – SOFTWARE BINARY ANALYSIS

5.1 OBJECTIVE

5.1.1 The objective of this activity is to determine if the firmware and companion mobile application of the Device Under Test (DUT) is free from:

- Common software errors such as buffer overflows;
- Known vulnerabilities in any of the third-party libraries being used; and
- Known Malware.

5.1.2 Devices that passes Assessment Tier 3 would likely be capable of resisting against script kiddies that leverages on readily available exploit kits.

5.2 REQUIREMENTS

5.2.1 The firmware and the companion mobile application shall be subjected to testing under automated binary analysers which shall be performed by a testing laboratory.

5.3 PROCESS

5.3.1 The developer shall provide the firmware binary and the companion mobile applications (if available) of the DUT to the testing laboratory.

5.3.2 To facilitate testing, the firmware binary and companion mobile applications must be provided in a format that is supported by the binary scanners (e.g. unencrypted, specific file extension, etc.). The developer shall exercise due diligence to scan and remove any malwares before submission.

5.3.3 The developer shall also provide a list of all software components (e.g. Micro_Httpd, OpenSSL, etc.) used in the DUT's firmware and companion mobile applications (iOS/Android), and state all permissions requested by the mobile applications (e.g. camera, location, Bluetooth, etc.).

5.3.4 In addition, the hash values (SHA-256) of all files submitted shall be provided.

5.3.5 On the receipt of the binary files, the testing laboratory shall proceed to perform the binary scans using a suite of binary analysis tools.

5.3.6 The generated binary analyser reports shall be analysed by the testing laboratory.

5.3.7 The required binary analysis tools are also available at the National Integrated Centre for Evaluation (NICE). For more information, please contact the CCC team.

5.4 SCOPE

5.4.1 The testing laboratory shall conduct the following tasks in around 3 – 5 working days, inclusive of submission of the full report.

Software Errors

5.4.2 Binary Code Analysis tool is used to identify common flaws such as buffer overflows. It is expected that there can be multiple false positives in the test results. The testing laboratory, together with the developer, is expected to evaluate all relevant findings.

5.4.3 For positive findings, the developer must apply remediation procedures. Following remediation procedures, the testing laboratory shall make re-test the binary code. The remediated findings and the remediation steps must be included in the report to CCC.

5.4.4 For each false positive, the testing laboratory must provide sufficient justification to explain why the finding is a false positive.

Vulnerabilities in third party libraries/components, and hard-coded sensitive security parameters

5.4.5 A Software Composition analyser is used to identify the usage of any third-party libraries and for such libraries, whether any known vulnerabilities (CVEs) are reported. The Software Composition analyser may also discover any hard-coded sensitive security parameters.

5.4.6 If the developer has successfully implemented the development process requirements specified in Tier 2, it is expected that the list of findings reported by the Software Composition analyser should be minimal.

5.4.7 Nonetheless, in some unexpected situations, the list of identified vulnerabilities might remain significant. For such situations, the developer is strongly encouraged to withdraw the application and focus on remediating the flaws, rather than incurring unnecessary cost to proceed with the application process.

5.4.8 Both the unfiltered (full list of identified vulnerabilities) and the filtered report will be used by the testing laboratory and the developer. The filtered report aims to assist the testing laboratory in prioritising the vulnerabilities to examine. Filtering is based on the following rules:

- Commonly used libraries that potentially have external interfaces that could be exploited;
- CVSS Attack Vector (AV): Network (N);
- CVSS Attack Complexity (AC): Low (L);
- CVSS Confidentiality (C): High/Low;
- CVSS Integrity (I): High/Low;

5.4.9 The testing laboratory shall assess that third-party libraries/components used by the firmware are compliant with respective license requirements (GNU General Public License, BSD license, MIT, Creative Commons, Apache, etc.).

5.4.10 The testing laboratory shall assess that there are no exploitable third-party libraries/components. In the event that vulnerabilities are deemed to be highly exploitable, the developer is required to update the libraries/components to a version without vulnerabilities, or to implement a custom patch/fix to address the vulnerability. The testing laboratory shall re-test the binary code following developer's remediation procedures. The remediated findings and its remediation steps must be included in the report to CCC.

5.4.11 The testing laboratory shall ensure that the firmware and the companion mobile application does not contain hard-coded critical security parameters.

5.4.12 For each false positive, the testing laboratory must work with the developer to provide sufficient justification on why the finding is a false positive.

Malware Scan

5.4.13 Developer shall ensure that the binary files submitted is free from known malware.

5.4.14 The binary files shall be subjected to a commercial malware scanner that exists as a cloud solution for malware analysis. Therefore, the developer shall consent to allowing the binary files to be uploaded to a commercial malware scanner for malware analysis.

5.4.15 In the event that firmware and/or the companion mobile application tests positive for malware, the initial malware scan results shall be confirmed using a different malware scanner. If both malware scanners confirm that the binary file tests positive for malware, CCC reserves the right to take appropriate actions against the developer.

Mobile Application Scan

5.4.16 Where a companion mobile app is available to facilitate the usage of the DUT, the companion mobile app shall be subjected to binary analysis. The testing laboratory shall prioritise their analysis of the companion mobile app on the following areas:

- Hardcoded credentials or critical security parameters;
- Exposure of sensitive information, for example via insecure storage or insecure communication channels;
- Potential intrusion to privacy for example whether the app requests for rights/permissions that it is deemed not to require such as to user's calendar or device's camera; or where data is sent out

despite the user explicitly denying such request.

5.4.17 Mobile applications across available platforms such as Android and iOS, as stated in the CLS application, shall be subjected to the binary analysis.

5.4.18 The findings shall be resolved or justified as appropriately.

Search for Vulnerabilities in the Public Domain

5.4.19 The testing laboratory shall examine sources of information publicly available to identify potential vulnerabilities in the DUT.

5.4.20 The testing laboratory shall also examine sources of information publicly available to identify generic vulnerabilities (vulnerabilities discovered on similar device-type) that could potentially be applicable for the DUT and determine if they are applicable for the DUT.

5.4.21 The testing laboratory can make use of several established sources. Examples are Common Vulnerabilities and Exposures (CVE), and public search engines (e.g. Google).

5.4.22 The testing laboratory shall also examine sources of information publicly available to check for DUT source code, unencrypted binary code, developer-confidential data, DUT user credentials, or other information that may be available to a potential attacker. E.g. source code or DUT default administrator credentials hosted on GitHub that are publicly accessible.

5.4.23 At this stage, the testing laboratory is not expected to conduct tests to verify if the identified vulnerabilities are exploitable.

5.5 PASS CRITERIA

5.5.1 The firmware and the companion mobile application shall be free from identified exploitable vulnerabilities using the binary analysers. For non-conformance, the developer and the testing laboratory can choose to provide due justification to CCC which must be supported by the testing laboratory. The exception will be reviewed and accepted by CCC on a case-by-case basis.

5.6 TESTING LABORATORY DELIVERABLES

5.6.1 The testing laboratory shall submit a report containing the following:

1. Verdict on the software errors
2. Verdict on the third-party library and hard-coded sensitive security parameters
3. Verdict on the mobile application scan (if applicable)
4. Results on the search for potential vulnerabilities in the public domain

5.6.2 If vulnerabilities are identified during testing, the testing laboratory shall describe the identified vulnerabilities in the report and state the method of resolution undertaken by the developer.

5.6.3 During the course of testing, if the testing laboratory discovers any discrepancies or false declarations in the developer's declaration of conformance to the Security Baseline Requirements or Lifecycle requirements, the testing laboratory is to provide the information to CCC, CCC reserves the full rights to enforce actions as described in Chapter 8.7 of CLS Publication #1 – Overview of the Scheme [3].

6 ASSESSMENT TIER #4 – BLACK BOX PENETRATION TESTING

6.1 OBJECTIVE

6.1.1 The objective of this activity is to determine if the DUT is resistant to the common IoT device attacks through black-box penetration testing.

6.1.2 Devices that passes Assessment Tier 4 should be capable of providing resistance against attacks conducted by a basic attacker on exposed interfaces.

6.1.3 The black box penetration test does not seek to assert that the DUT is resistant to all attacks.

6.1.4 However, the penetration test should provide basic assurance that the DUT is adequate to ward off the commonly known and straightforward attacks against such devices.

6.2 PRE-REQUISITES

6.2.1 The developer shall provide the following to the testing laboratory:
1. Guidance document (installation/operation guide)
2. Sufficient number of DUT to meet testing laboratory's requirements

6.3 SCOPE

6.3.1 This activity comprises the following tasks:

No.	Tasks
1	Device setup and verification of guidance documents
2	ESTI Conformance Verification - verifying that the device indeed implemented the security measures that the developer has declared and specified in the checklist.
3	Scheme-mandated minimum test specifications
4	Search for potential vulnerabilities in the public domain
5	Vulnerability analysis and freeform penetration testing, devising test cases based on: a) The report from Assessment Tier #3; b) Known threat vectors; c) The laboratory's expertise and experience.
6	Password cracking (if applicable)

Table 3 – Assessment Tier #4 tasks

6.3.2 The testing laboratory shall conduct the abovementioned tasks concurrently where possible by leveraging on multiple units of the device and it is expected that it should take no longer than 15 working days, inclusive of drafting the test report.

6.3.3 Nonetheless, the testing laboratory is required to spend a minimum of 4

days on Freeform Penetration Testing. The objective of this freeform testing is to serve as a feedback loop for the continuous refinement of the minimum test specification so to align with the current threat landscape.

6.3.4 The developer shall facilitate the testing by the testing laboratory. For example, by providing sufficient units of the devices to the testing laboratory and responding to queries. The developer shall note that certain tests might render the device to be unusable (e.g. physically damaged).

Device setup and verification of guidance documents

6.3.5 The objective of analysing the guidance document provided alongside the DUT is to ensure that the user guidance does not mislead the user into installing or operating the DUT in an insecure manner, and to minimise the risk of human or other errors in operation that may affect the security of the DUT.

6.3.6 The guidance document (i.e. user manual, installation guide, operation guide, etc.) shall consist of clear steps that guides the end-user to install and operate the DUT in a secure manner. The guidance document shall be written in a manner that is easily understood by the typical user of the DUT. As an example, for a smart home appliance, it can be assumed that the typical user has little to no knowledge of cybersecurity. If the DUT functions are configurable, the guidance document shall indicate secure values as appropriate. The guidance document shall also describe possible modes of operation of the DUT, their consequences and procedures for returning the DUT back into a secure configuration.

6.3.7 The testing laboratory shall examine the guidance document(s) provided to ensure that the guidance document provided meets the requirements stated above.

ESTI Conformance Verification

6.3.8 As part of the application, the developer is required to declare against the provisions specified in the checklist and provide evidence and descriptions of how these requirements have been implemented by the device.

6.3.9 The testing laboratory examines that these security measures are indeed being implemented and that such implementation are appropriate to fulfil to the requirements.

Scheme-mandated Minimum Test Specifications

6.3.10 In order to ensure consistent penetration testing of connected products across different testing laboratories, minimum test specifications for the different categories of connected products are defined.

6.3.11 The testing laboratory shall ensure that the test objectives in the test specifications are achieved prior to the conduct of independent vulnerability analysis and penetration testing.

6.3.12 The testing laboratory shall take reference from CLS Publication – Minimum Test Specifications and Methodology for Tier 4 [4] for this task.

6.3.13 It is of CCC's intention that the test specifications shall be revised in the future to keep up with the evolving threat landscape.

Search for potential vulnerabilities in the public domain

6.3.14 The testing laboratory shall examine sources of information publicly available to identify potential vulnerabilities for the DUT.

6.3.15 The testing laboratory shall also examine sources of information publicly available to identify generic vulnerabilities (vulnerabilities discovered on similar DUT-type) that could potentially be applicable for the DUT and determine if they are applicable for the DUT.

6.3.16 The testing laboratory can make use of several established sources. Examples are Common Vulnerabilities and Exposures (CVE), and public search engines (e.g. Google).

6.3.17 The testing laboratory shall also examine sources of information publicly available to check for DUT source code, binary code, developer-confidential data, DUT user credentials, or other information that may be available to a potential attacker. E.g. source code or DUT default administrator credentials hosted on GitHub.

Vulnerability Analysis

6.3.18 From information collected through the preceding search for potential vulnerabilities in the public domain and from the report of the binary analysis covered under Tier 3, the developer shall devise a list of potential security vulnerabilities and potential attack paths.

6.3.19 The testing laboratory may make use of vulnerability scanning tools and techniques to identify potential vulnerabilities.

6.3.20 Malformed Input Testing (also known as fuzz testing) should be conducted to discover coding errors, security loopholes in the software of the DUT. It involves inputting massive amounts of random data to the DUT in an attempt to make it malfunction and discover potential flaws.

6.3.21 The testing laboratory shall make use of automated fuzzing software tools. Due to the limited time period, it is advised that the testing laboratory focus time and effort on interfaces that are deemed more critical.

6.3.22 It is expected that fuzz testing may result in device crashes which is different from an exploitable vulnerability. The developer, together with the testing laboratory, shall to their best effort, attempt to perform analysis on the crashes to determine if the issues are potentially an exploitable vulnerability.

6.3.23 When devising attack scenarios, the operational environment in which the DUT is expected to be used should be taken into consideration. For example, smart home devices are usually placed in the home and thus are not subjected to attackers with physical access to visible interfaces. Attacks are usually conducted through the network that the smart devices are connected to. The attack scenarios shall focus on the logical interfaces accessible by potential attackers. On the other hand, a smart door lock that is installed in publicly accessible locations might be subjected to simple non-destructive physical tests.

6.3.24 The testing laboratory should identify sensitive assets that must be protected and devise attack scenarios to test that the sensitive assets are indeed adequately protected (e.g. Sensitive and private user data must be encrypted, cryptographic keys, passwords etc.).

Penetration Testing

6.3.25 The testing laboratory shall prioritise the test cases to ensure the intended outcome of the labelling scheme could be achieved.

6.3.26 The testing laboratory is not expected to perform advanced attacks (e.g. laser injection, hardware side channel attacks). However, should such attacks be feasible within the timeframe of the testing or be practically executed by a potential attacker in the actual deployment environment, the testing laboratory shall execute such attacks on the DUT during testing.

Password Cracking

6.3.27 If the testing laboratory manages to obtain encrypted files containing sensitive credentials (user credentials, credentials to associated web services, etc.), the testing laboratory shall explore the brute-forcing of these files in an attempt to retrieve them.

6.4 PASS CRITERIA

6.4.1 The DUT is deemed pass if no critical or significant vulnerabilities are uncovered.

6.5 DELIVERABLES

6.5.1 The testing laboratory shall submit a concise test report containing the following:

1. Executive Summary
2. Verdict on the analysis of guidance document
3. Test results from tests in Minimum Test Specification.
    a. For test cases the DUT passes, an indicative statement by the lab would suffice.
    b. For test cases which the DUT failed, the lab shall record the

detailed setup and procedure such that the results could be reproduced.

4. Results on the search for potential vulnerabilities in the public domain, including the list of search terms.
5. Test cases and results of the penetration testing. The test cases could be described in high level. Recording of detailed setup and procedures are required only for test cases which succeeded in exploiting the DUT.

6.5.2 The testing laboratory shall also arrange for a meeting with CCC to present the results.

6.5.3 The testing laboratory may be required to perform additional testing if CCC deems the testing performed to be inadequate.

6.5.4 During the course of testing, if the testing laboratory discovers any discrepancies or false declarations in the developer's declaration of conformance to the Security Baseline Requirements or Lifecycle requirements, the testing laboratory is to provide the information to CCC, CCC reserves the full rights to enforce actions as described in Chapter 8.7 of CLS Publication #1 – Overview of the Scheme [3].

7 CONFORMANCE CHECKLIST

7.1.1 The checklist defines the provisions that shall be met at each tier of the CLS. The requirements (and corresponding checklist) may vary from time-to-time. Developers are encouraged to refer to the latest checklist before applying.

7.1.2 The checklist is intended to be used in tandem with ETSI EN 303 645 – Cyber Security for Consumer Internet of Things [1] and IMDA IoT Cyber Security Guide [2] published by IMDA. Please refer to the respective documents for the detailed description of the provisions.

7.1.3 The provisions (5.1-1 to 5.13-1, 6-1 to 6-5) within this document are reproduced from the ETSI EN 303 645 © European Telecommunications Standards Institute 2020. Further use, modification, copy and/or distribution are strictly prohibited.

7.1.4 The mandatory clauses for each CLS Level are marked in green.

7.1.5 The developer is required to complete and submit the following checklist for all levels for CLS. The developer is required to declare against **ALL** clauses even if the clauses may not be mandatory for the level the developer is applying. “M” refers to Mandatory, whereas “R” refers to “Recommended”. “C” refers to “Conditional” should a dependent provision is being implemented.

7.1.6 The checklist states the required supporting evidence (to show how the developer fulfils the respective provisions) under the ‘Description of how the provision is fulfilled’ column. Depending on the provisions, the developer shall provide supporting evidences which can include the following:

- Process-related provisions: Quality manual, process documents, work instructions, checklist, and policy documents
- Technical provisions: Technical/design overview/specifications/diagrams, accompanying user guidance documents, user interface screenshots that helps to depict the implemented technical requirements

7.1.7 The developer is required to note down/state the page number of the content that fulfils the respective points within the provision.

Clause	Provision	CLS Requirements				Developer's Conformance (Yes/No/Not Applicable)	Description of how the provision is fulfilled.
		L1	L2	L3	L4		
The following are the 14 provisions from the ETSI EN 303 645.							
5.1: No universal default passwords	**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.	M C (1)	M C (1)	M C (1)	M C (1)		Supporting evidence shall show how it is ensured that passwords are unique per device. 1. If pre-installed passwords are used, the same universal default values should not be used across devices. 2. The device must require that the user define a new password during initialisation. Please note that there are many mechanisms used for performing authentication, and passwords are not the only mechanism for authenticating a user to a device. If other authentication mechanisms are used, please provide details.
	5.1-2: Where pre-installed unique per device passwords are	M C (2)	M C (2)	M C (2)	M C (2)		Supporting evidence shall describe the following:

	used, these shall be generated with a mechanism that reduces the risk of automated attacks against a class or type of device.						1. How the pre-installed passwords are generated for each device and what is done to ensure that the pre-installed passwords are sufficiently random. 2. Where and when are the passwords generated (e.g. off-device and provisioned onto the device subsequently, or generated upon device's initial boot-up sequence)? 3. How are the randomised passwords generated? Was a random function or a cryptographically secure pseudo random number generator used? Are the randomised passwords based on any device information (MAC address, etc.)? Minimally, the following are required for pre-installed passwords: 1. Passwords with incremental counters ("password1", "password2") are not allowed. 2. Pre-installed passwords must

							be sufficiently randomised using a random function. 3. Passwords must not be relatable in an obvious manner to public information such as MAC address or Wi-Fi SSID. The developer shall also provide 5 instances of randomised passwords that are generated using the aforementioned password randomisation mechanism to CCC. Please note that there are many mechanisms used for performing authentication, and passwords are not the only mechanism for authenticating a user to a device. If other authentication mechanisms are used, please provide details.
	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.	M	M	M	M		Supporting evidence shall list all authentication mechanisms (e.g. passwords, tokens, smart cards, digital signatures, biometrics, etc.) available for the various device login-interfaces (e.g. device configuration portal, companion mobile application, etc.), and describe how each of the mechanisms are adequately secured to address the risk

							and usage scenario, and the best practice cryptography that were referenced (if used).
	5.1-4: 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.	M C (8)	M C (8)	M C (8)	M C (8)		Supporting evidence shall show the password reset/change mechanism(s) that the consumer may use to change the authentication value.
	5.1-5: When the device is not a constrained device, it shall have a mechanism available which makes bruteforce attacks on authentication mechanisms via network interfaces impracticable.	M C (5)	M C (5)	M C (5)	M C (5)		Supporting evidence shall describe the employed authentication rate limiting policy for making brute force attacks impracticable on each of the device's login-interfaces. Examples of login-interfaces not limited to the following: • Device and/or device management portal login; • Companion Mobile Application login; • Other network interfaces, ports or services. For each of the login-interfaces available on the device, supporting evidence shall describe the following:

							1. What is the maximum number of attempts within a certain time interval? 2. What happens when a certain number of failed authentication attempts is reached? Minimally, for each of the device's login-interfaces, the device shall employ a rate-limiting mechanism that has a limitation on the number of authentication attempts within a certain time interval, and locks/delays additional authentication attempts after a limited number of failed authentication attempts.
5.2: Implement a means to manage reports of vulnerabilities	**5.2-1:** The manufacturer shall make a vulnerability disclosure policy publicly available. This policy shall include, at a minimum: • contact information for the reporting of issues; and • information on timelines for:	M	M	M	M		Supporting evidence shall describe the following: 1. Contact information for the reporting of issues by listing down all various contact mechanisms available for the public to report vulnerabilities, and where information of each of the contact mechanisms are located. E.g. Contact numbers and/or email address are listed on developer's

	1) initial acknowledgement of receipt; and 2) status updates until the resolution of the reported issues.						website and user guidance documents; Use of a web form; Use of a vulnerability coordination and bug bounty platform (e.g. HackerOne). 2. Procedures around the initial acknowledgement of receipt. 3. Procedures around the status updates of the vulnerability until it is resolved. 4. [For 5.2-2] The internal guidelines/policies describing the expected time required for resolving vulnerabilities. 5. [For 5.2-3] Supporting evidence that describe internal processes for continuous monitoring, identification, and rectification of security vulnerabilities.
	5.2-2: Disclosed vulnerabilities should be acted on in a timely manner.	R	R	R	R		
	5.2-3: Manufacturers should continually monitor for, identify and rectify security vulnerabilities within products and services they sell, produce, have produced and services they operate during the defined support period.	R	R	R	R		
5.3: Keep software updated	**5.3-1:** All software components in consumer IoT devices should be securely updateable.	R	R	R	R		Supporting evidence shall list all the software components in the device and describe how each of them can be securely updateable.
	5.3-2: When the device is not a	M C (5)	M C (5)	M C (5)	M C (5)		Supporting evidence shall describe the various update mechanisms supported

	constrained device, it shall have an update mechanism for the secure installation of updates.						by the device and how each of them are secure. The update mechanism(s) should ensure the authenticity and integrity of software updates. Examples of secure update mechanisms not limited to the following: • Updates to be transferred over a secure channel (HTTPS); • Device should employ anti-rollback policy based on version checking of the firmware; • For updates that are downloaded manually from the developer's website by the user, the firmware is encrypted and there is a mechanism for the end-user to verify the authenticity and integrity of the firmware. Some devices may not be able to support or be required to provide software updates. For such constrained devices, this provision may not be applicable. Please provide justification for how the device is a constrained device.
	5.3-3: An update shall be simple	M C (12)	M C (12)	M C (12)	M C (12)		Supporting evidence shall describe the various software update mechanisms

	for the user to apply.						available. Some devices may not be able to support or be required to provide software updates. For such constrained devices, this provision may not be applicable. Please provide justification for how the device is a constrained device.
	5.3-4: Automatic mechanisms should be used for software updates.	R C (12)	R C (12)	R C (12)	R C (12)		Supporting evidence shall describe the automatic software update mechanism available on the device. Examples of automatic software update mechanisms, not limited to the following: • The device is to download the update automatically and installs the update at either a stipulated timing or when the device is restarted. • The device is updated automatically via the companion mobile application.
	5.3-5: The device should check after initialization, and then periodically, whether security updates are available.	R C (12)	R C (12)	R C (12)	R C (12)		Supporting evidence shall describe the schedule and/or frequency of checks made by the device for available security updates.

	5.3-6: If the device supports automatic updates and/or update notifications, these should be enabled in the initialized state and configurable so that the user can enable, disable, or postpone installation of security updates and/or update notifications.	R C (9, 12)	R C (9, 12)	R C (9, 12)	R C (9, 12)		Supporting evidence shall show the following: 1. the update notifications to the user 2. the default configuration for such notifications 3. the available options that can be taken by the user when security updates are available
	5.3-7: The device shall use best practice cryptography to facilitate secure update mechanisms.	M C (12)	M C (12)	M C (12)	M C (12)		Supporting evidence shall state the cryptographic functions and algorithms used, and/or referenced standards (if any) to support the secure update mechanism(s) as stated in Provision 5.3-2. Examples of best practice cryptography for secure update mechanisms, not limited to the following: • Use of TLS 1.2 and above for the communication of security updates • Use of a digital signature for verifying the authenticity and integrity of the software updates
	5.3-8:	M C	M C	M C	M C		Supporting evidence shall describe the

	Security updates shall be timely.	(12)	(12)	(12)	(12)		internal policies on ensuring the availability of security updates in a timely manner.
	5.3-9: The device should verify the authenticity and integrity of software updates.	R C (12)	R C (12)	R C (12)	R C (12)		Refer to supporting evidence requirements from ETSI 5.3-2 and 5-3-7.
	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.	M (11, 12)	M (11, 12)	M (11, 12)	M (11, 12)		
	5.3-11: The manufacturer should inform the user in a recognizable and apparent manner that a security update is required together with information on the risks mitigated by that update.	R C (12)	R C (12)	R C (12)	R C (12)		Supporting evidence shall describe the notification mechanism(s) in which the user is informed of a security update.
	5.3-12: The device should notify the user when the application of a software update will disrupt the basic functioning of the device.	R C (12)	R C (12)	R C (12)	R C (12)		Supporting evidence shall show how the user is informed of a software update that will disrupt the basic functioning of the device.

	5.3-13: The manufacturer shall publish, in an accessible way that is clear and transparent to the user, the defined support period.	M	M	M	M		Supporting evidence shall list all avenues in which information on the defined support period is provided to the consumer (e.g. website, product retail packaging, user guidance document, etc.). Actual documents or screenshots of each of the avenue shall be included. Minimally, the defined support period must be provided on the developer's website.
	5.3-14: For constrained devices that cannot have their software updated, the rationale for 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.	R C (3, 4)	R C (3, 4)	R C (3, 4)	R C (3, 4)		For constrained devices that cannot be updated, supporting evidence shall describe how the user can be informed of security vulnerabilities, and the planned approach/resolution method (e.g. hardware replacement, etc.) that would be available to the user.
	5.3-15: For constrained devices that cannot have their software updated, the product should be isolable	R C (3, 4)	R C (3, 4)	R C (3, 4)	R C (3, 4)		

	and the hardware replaceable.						
	5.3-16: The model designation of the consumer IoT device shall be clearly recognizable, either by labelling on the device or via a physical interface.	M	M	M	M		Supporting evidence shall state where the consumer can find the model designation of the consumer IoT device.
5.4: Securely store sensitive security parameters	**5.4-1:** Sensitive security parameters in persistent storage shall be stored securely by the device.	R	R	R	M		Supporting evidence (technical specifications, security architecture, key lifecycle diagrams, etc.) shall describe how the sensitive security parameters are stored and communicated securely. Please also state and list down all sensitive security parameters, hard-coded unique per device identities that are available on the device (stored in firmware, secure storage mechanisms, use of a certified IoT platform, etc.), and the secure storage mechanism used for each of them.
	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.	R C (10)	R C (10)	R C (10)	M C (10)		
	5.4-3: Hard-coded critical security parameters in device software source code shall not be used.	R	R	M	M		
	5.4-4: Any critical security	R	R	M	M		

	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.						
5.5: Communicate securely	**5.5-1:** The consumer IoT device shall use best practice cryptography to communicate securely.	R	R	R	M		Supporting evidence shall describe how the device communicates securely and state the best practice cryptography standards/guidelines that were referenced (if used).
	5.5-2: The consumer IoT device should use reviewed or evaluated implementations to deliver network and security functionalities, particularly in the field of cryptography.	R	R	R	R		Supporting evidence shall list all reviewed or evaluated cryptographic implementations used for network and security functionalities.
	5.5-3: Cryptographic algorithms and primitives should be updateable.	R	R	R	R		Supporting evidence shall describe how the cryptographic algorithms and primitives used are updateable (e.g., the cryptographic library and algorithms

							used could be updated with a software update).
	5.5-4: Access to device functionality via a network interface in the initialized state should only be possible after authentication on that interface.	R	R	R	R		Supporting evidence shall list all network interfaces on the device and the authentication mechanisms available on all of the network interfaces. In addition, if certain device functionalities are available prior to authentication, a description of the purpose for allowing those functionalities shall be provided.
	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.	R	R	R	M		Supporting evidence shall describe that authentication is required prior to making security-relevant changes. Example scenarios: • Administrator's authentication is required prior to making changes in the device's web configuration portal. • Administrator's authentication is required prior to configuring security-relevant changes to the device using the companion mobile application. • Authentication should also be required for any other interfaces/methods that facilitates making security-relevant changes.

	5.5-6: Critical security parameters should be encrypted in transit, with such encryption appropriate to the properties of the technology, risk and usage.	R	R	R	R		Supporting evidence shall list all critical security parameters that are communicated across devices, associated services, or companion mobile applications, and describe the encryption used to protect them during transit.
	5.5-7: The consumer IoT device shall protect the confidentiality of critical security parameters that are communicated via remotely accessible network interfaces.	R	R	R	M		Supporting evidence that describes the critical security parameters that are communicated via remotely accessible network interfaces and the mechanism used to protect them.
	5-5.8: The manufacturer shall follow secure management processes for critical security parameters that relate to the device.	R	M	M	M		Supporting evidence (e.g. key lifecycle diagram) shall describe the lifecycle (creation, provisioning, renewal, revocation) of critical security parameters. Please state referenced standards/best practices for secure management processes, if used.
5.6: Minimise exposed attack surfaces	**5.6-1:** All unused network and logical interfaces shall be disabled.	R	R	R	M		Supporting evidence shall list all network and logical interfaces that are currently enabled in the device's default configuration and provide a description for their functionality and purpose. If a firewall is available on the device, the

							firewall rules should also be provided.
	5.6-2: In the initialized state, the network interfaces of the device shall minimize the unauthenticated disclosure of security-relevant information.	R	R	R	M		Supporting evidence shall describe the measures taken to ensure that security-relevant information is not exposed via the network interfaces of the device.
	5.6-3: Device hardware should not unnecessarily expose physical interfaces to attack.	R	R	R	R		Supporting evidence shall list all physical interfaces available on the device and describe any measures (if necessary) taken to secure physical interface(s) to fulfil this provision.
	5.6-4: Where a debug interface is physically accessible, it shall be disabled in software.	R C (13)	R C (13)	R C (13)	M C (13)		Supporting evidence shall list all available hardware debug interfaces available and describe the steps taken in ensuring that they are disabled.
	5.6-5: The manufacturer should only enable software services that are used or required for the intended use or operation of the device.	R	R	R	R		Supporting evidence shall describe the software services available on the device and their status (enabled/disabled), along with a description of the rationale behind enabling and disabling each of these services.
	5.6-6: Code should be minimized to the functionality necessary for the service/device to operate.	R	R	R	R		Supporting evidence shall list all software code and libraries available on the device and describe what has been done to minimise the existence of unused code.

	5.6-7: Software should run with least necessary privileges, taking account of both security and functionality.	R	R	R	R		Supporting evidence shall list the software services that are running on the device, and the corresponding privileges assigned to each of them, along with a description of the rationale behind the assignment of the level of privileges to each of these services.
	5.6-8: The device should include a hardware-level access control mechanism for memory.	R	R	R	R		Supporting evidence shall describe the hardware-level access control mechanism for memory employed.
	5.6-9: The manufacturer should follow secure development processes for software deployed on the device.	R	R	R	R		Please refer to clause CK-LP-02.
5.7: Ensure software integrity	**5.7-1:** The consumer IoT device should verify its software using secure boot mechanisms.	R	R	R	R		Supporting evidence shall describe the secure boot mechanism used (use of certified IoT platform, or the hardware root of trust utilised, the secure boot process), and the device's behaviour following the detection of unauthorised changes to its software.
	5.7-2: 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 than those necessary to	R	R	R	R		

	perform the alerting function.						
5.8: Ensure that personal data is protected	**5.8-1:** The confidentiality of personal data transiting between a device and a service, especially associated services, should be protected, with best practice cryptography.	R	R	R	R		Supporting evidence shall list the following: 1. All sensitive personal data communicated between the device and associated services, and describe how they are adequately secured to address the risk and usage scenario, and the best practice cryptography that were referenced (if used). 2. All external sensing capabilities available on the device and state where this information is provided to the user.
	5.8-2: 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.	R	M	M	M		
	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.	R	M	M	M		
5.9: Make systems resilient to	**5.9-1:** Resilience should be built in to consumer IoT devices	R	R	R	R		Supporting evidence shall describe how these provisions are fulfilled.

outages	and services, taking into account the possibility of outages of data networks and power.						
	5.9-2: 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.	R	R	R	R		
	5.9-3: 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.	R	R	R	R		
5.10: Monitor system telemetry data	**5.10-1:** If telemetry data is collected from consumer IoT devices and services, such as usage and measurement data, it should be examined for security anomalies.	R C (6)	R C (6)	R C (6)	R C (6)		Supporting evidence shall list all telemetry data collected and provide the purpose for the collection of each of them.

5.11: Make it easy for consumers to delete personal data	**5.11-1:** The user shall be provided with functionality such that user data can be erased from the device in a simple manner.	R	M	M	M		Supporting evidence shall show the functionality that allows user data to be erased from the device, associated services, and companion mobile application.
	5.11-2: The consumer should be provided with functionality on the device such that personal data can be removed from associated services in a simple manner.	R	R	R	R		
	5.11-3: Users should be given clear instructions on how to delete their personal data.	R	R	R	R		
	5.11-4: Users should be provided with clear confirmation that personal data has been deleted from services, devices and applications.	R	R	R	R		
5.12: Make installation and maintenance of devices easy	**5.12-1:** Installation and maintenance of consumer IoT should involve minimal decisions by the user and should follow security best	R	R	R	R		Supporting evidence shall show the following: 1. How the device is already pre-configured or presents the configuration options with the

	practice on usability.						most appropriate security options turned on/selected. 2. Specific steps that guide users to securely set up the device. 3. Specific steps that guide users on checking whether the device is securely set up.
	5.12-2: The manufacturer should provide users with guidance on how to securely set up their device.	R	R	R	R		
	5.12-3: The manufacturer should provide users with guidance on how to check whether their device is securely set up.	R	R	R	R		
5.13: Validate input data	**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.	R	R	M	M		Supporting evidence shall describe the input validation strategies employed for the following data-input interfaces: 1. Authentication mechanism (device management portal, companion mobile application, etc.) 2. All data-input text boxes within the various user-interfaces (device management portal, companion mobile applications, etc.). 3. All APIs used between networks in services and devices. Server to

							server (backend) connections and services are considered to be out of scope.
6: Data protection provisions for consumer IoT	**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.	R	M	M	M		Supporting evidence shall describe how each of the provisions are fulfilled and show the following: 1. The mechanism(s) in which the user is provided with clear and transparent information on the processing of personal data and/or collection/processing of telemetry data. 2. The mechanism(s) used to obtain the user's consent on the processing of personal data. 3. The mechanism(s) available for user to withdraw consent for the processing of personal data.
	6.2: Where personal data is processed on the basis of consumers' consent, this consent shall be obtained in a valid way.	R C (7)	M C (7)	M C (7)	M C (7)		
	6.3: Consumers who gave consent for the processing of their personal data shall have the capability to withdraw it at any time.	R	M	M	M		
	6.4: If telemetry data is	R C (6)	R C (6)	R C (6)	R C (6)		

	collected from consumer IoT devices and services, the processing of personal data should be kept to the minimum necessary for the intended functionality.						
	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.	R C (6)	M C (6)	M C (6)	M C (6)		
Conditions: 1) Passwords are used; 2) Pre-installed passwords are used; 3) Software components are not updateable; 4) The device is constrained; 5) The device is not constrained; 6) Telemetry data being collected; 7) Personal data is processed on the basis of consumers' consent; 8) The device allowing user authentication; 9) The device supports automatic updates and/or update notifications; 10) A hard-coded unique per device identity is used for security purposes; 11) Updates are delivered over a network interface; 12) An update mechanism is implemented; 13) A debug interface is physically accessible							

The following are the lifecycle security considerations of the IMDA IoT Cyber Security Guide.							
CK-LP-01	Have you conducted threat modelling to identify, analyse and mitigate threats to the device?	-	M	M	M		Provide internal document(s) which defines the process of threat modelling including: • Identify the potential target(s)/assets/ areas of interest to be protected • Define the security problem • Conduct risk assessment • Determine the security objectives • Define the security requirements • Design and implement • Validate and verify that the capabilities address the security requirements
CK-LP-02	Did you design and develop the device using a secure engineering approach?	-	M	M	M		Provide supporting document(s) to provide confidence that secure engineering approaches have been adopted and are effective. Examples include the following: • Reuse existing, well-secured software: evidence showing the code repository used to store and maintain secured software for reuse when suitable, or internal documents describing the process for the storage and usage of

							secured software. • Secure coding practices: internal documents describing the process to ensure secure coding practices are followed during the development of the device. List the standard, guideline, security best practices that are referenced. • Improve executable security: evidence showing compiler and build tools configuration, or internal documents describing the process to improve the executable security. • Functional testing of security features: test document such as functional testing test case document or test tool report describing the test cases (purpose and steps of each test case), or internal document(s) describing the process to conduct functional testing. • Developer and/or peer code review: Internal document or evidence of the tracking system used for tracking the code review feedback and remediation status of the findings. • Static application security testing (SAST): test report describing the

							result of SAST test or internal document(s) describing the process of SAST. • Dynamic analysis security testing (DAST): tool report describing the result of DAST test or internal document(s) describing the process of DAST. • Application programming interfaces (API) testing: tool report describes the result of API test or internal document(s) describing the process of API testing. • Fuzz testing: tool report describing the result of fuzzing or internal document(s) describing the process of fuzz testing. Indicate the tools used to conduct the above test and state other integrated security related activities conducted (if any).
CK-LP-03	Do you implement and maintain the device with components from a secure supply chain, with no known unmitigated vulnerabilities?	-	M	M	M		Provide internal document(s) showing the following measures are conducted to ensure the device components have no known unmitigated vulnerabilities: • Patching all vulnerable third-party libraries that are used (e.g. OpenSSL, underlying Linux etc)

							and remove older versions that are no longer being used. • Defining the criteria for evaluation, selection, monitoring of performance and re-evaluation of supplier.
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?	-	M	M	M		Provide internal document(s) or evidence showing that the following security information are provided in simple language and timely manner: • Security policies. • End-of-life notifications.
CK-LP-05	Do you ensure that the device is hardened prior to release?	-	M	M	M		Provide internal document(s) that describes at least one measure on how device hardening is done. Examples: • Remove all backdoors. • Remove all debug codes from the released version. • Change default configuration and disable unnecessary services.
CK-LP-06	Do you maintain an inventory of components including its version, applied patches and updates?	-	M	M	M		Provide internal document(s) or evidence showing: • Software Build of Material (BOM) • Hardware BOM • Mobile application BOM

							• Version control system such as Subversion, Git and etc.
CK-LP-07	Do you conduct penetration testing and/or vulnerability assessment periodically, and before each major release?	-	M	M	M		Provide internal document(s) that describes the process of penetration testing (conducted by either internal penetration testing team or external vendors) and/or vulnerability assessment and the test tool(s) used.
CK-LP-08	Do you establish proper vulnerability disclosure and management?	-	M	M	M		Provide internal document(s) describing the following processes: • Supply chain capability to ensure upgrades and patches is provided. • Change management processes to manage security patch or updates.
CK-LP-09	Do you ensure that identities, certificates and secrets are secured throughout the lifecycle (e.g. creation, provisioning, renewal and revocation)?	-	M	M	M		Refer to ETSI EN 303 645 Provision 5.5-8

Table 4 – CLS Requirements checklist

8 REFERENCES

[1] ETSI, “Cyber Security for Consumer Internet of Things,” ETSI EN 303 645.

[2] Info-communications Media Development Authority of Singapore, “IMDA Internet of Things (IoT) Cyber Security Guide”.

[3] C. S. A. o. Singapore, “CLS Publication #1 - Overview of CLS,” Version 1.0, October 2020.

[4] C. S. A. o. Singapore, “CLS Publication - Minimum Test Specifications and Methodology for Tier 4,” Version 1.0, October 2020.

9 ACRONYMS

The following acronyms are used in CLS Publication 1, 2 and 3:

CC	Common Criteria for Information Technology Security Evaluation
CCC	Cybersecurity Certification Centre
CCTL	Common Criteria Testing Laboratories
CLS	Cybersecurity Labelling Scheme
CSA	Cyber Security Agency of Singapore
DUT	Device Under Test
ETSI	European Telecommunications Standards Institute
HPL	Historical Product List
IMDA	Info-communications Media Development Authority
IoT	Internet of Things
LPL	Labelled Product List
SCCS	Singapore Common Criteria Scheme
TL	Testing Laboratory