CCC SP-151-2

Cybersecurity Labelling Scheme

BY CYBER SECURITY AGENCY OF SINGAPORE

Cybersecurity Labelling Scheme for IoT Publication No. 2

Scheme Specifications

September 2023
Version 1.3

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	October 2020	Cyber Security Agency of Singapore	Release
1.1	April 2021	Cyber Security Agency of Singapore	Minor editorial revisions
1.2	October 2022	Cyber Security Agency of Singapore	Revised framework
1.3	September 2023	Cyber Security Agency of Singapore	Revised Process

CONTENTS

NOTICE

The Cyber Security Agency of Singapore makes no warranty of any kind with regard to this material and shall not be liable for errors contained herein or for incidental or consequential damages in connection with the use of this material.

INTRODUCTION

1.0.1 This document aims to provide an overview of Cybersecurity Labelling Scheme for IoT [CLS(IoT)] scheme. It outlines the four (4) cybersecurity levels, the assurance activities, acceptance criteria, and the expected deliverables of each of the levels.

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

1.0.3 This document is organised in the following manner:

a. Chapter 1 provides a broad overview of the 4 cybersecurity levels required under the different labelling levels of the CLS(IoT).

b. Chapter 2 elaborates on Level 1 – Declaration of Conformity to Security Baseline Requirements. It lists the objective, requirements, and the acceptance criteria.

c. Chapter 3 elaborates on Level 2 – Declaration of Conformity to International Standards. It lists the objective, requirements, and the acceptance criteria.

d. Chapter 4 elaborates on Level 3 – Declaration of Conformity to International Standards and Lifecycle Requirements, and Software Binary Analysis. It lists the requirements, test scope, pass criteria, and the test deliverables expected by CCC.

e. Chapter 5 elaborates on Level 4 – Declaration of Conformity to International Standards and Lifecycle Requirements, Software Binary Analysis, and Penetration Testing. It lists the requirements, test scope, pass criteria, and the test deliverables expected by CCC.

.

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 Assessments covered in CLS(IoT).
3. **Cybersecurity Certification Centre (CCC)** that oversees the CLS(IoT).

1.0.5 The CLS(IoT) 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 organisational 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.

1 OVERVIEW

1.1 CYBERSECURITY LABELING SCHEME FOR IOT [CLS(IOT)]

1.1.1 The following table provides an overview of the broad requirements for each labelling level of the CLS(IoT).

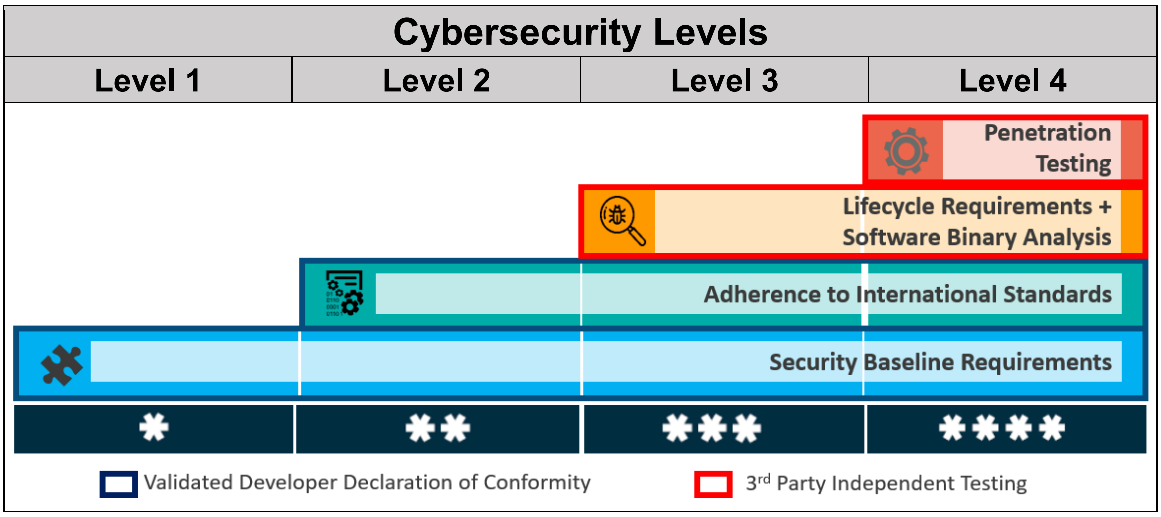
Cybersecurity Levels
Level 1
Level 2
Level 3
Level 4
Penetration Testing
Lifecycle Requirements + Software Binary Analysis
Adherence to International Standards
Security Baseline Requirements
✱
✱✱
✱✱✱
✱✱✱✱
Validated Developer Declaration of Conformity
3rd Party Independent Testing

Table 1 – CLS(IoT) Overview

2 LEVEL 1 – SECURITY BASELINE REQUIREMENTS

2.1 OBJECTIVE

2.1.1 The objective of Level 1 is to ensure that the Device Under Test (DUT) conforms to a minimal set of security baseline requirements.

2.1.2 **Level 1 is based solely on <u>declaration of conformity</u> by the developer**.

2.1.3 Devices that have completed Level 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.

2.2 REQUIREMENTS

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

2.2.2 The developer shall conform to the following mandatory cyber security provisions:

- No universal default passwords.
- Implement a means to manage reports of vulnerabilities.
- Keep software updated.

2.3 DECLARATION OF CONFORMITY

2.3.1 Developers are required to complete and submit the following to declare conformity to the security requirements:

- Declaration of Conformity.
- Required supporting evidence.

2.3.2 The manufacturer shall refer to the “CLS(IoT) Publication #4 – Assessment Methodology” for details on the minimum requirement and the expected supporting evidence required for each provision. Some examples of supporting evidence include detailed descriptions, screenshots, process charts, work instructions. This expected supporting evidence are listed in the Supporting Evidence template.

2.3.3 The supporting evidence can be provided in the following forms:

- Provided in the entirety of the related documentation, with specific reference to the actual chapter/section/paragraph which contains the required supporting evidence to substantiate the claim to the meeting of the requirement.
- Provided in forms of screen captures or snippets of the actual document where the snippets shall contain the required supporting evidence to substantiate the claim to the meeting of the requirement.

2.4 ACCEPTANCE CRITERIA

2.4.1 No independent testing by the testing laboratory is required for this level.

2.4.2 However, CCC or the approved CLS(IoT) TL will review the submitted Declaration of Conformity document and Supporting Evidence template. Level 1 is only considered satisfied when CCC gains assurance through the submitted supporting evidence that the requirements are met.

2.4.3 The CLS(IoT) label is awarded by CCC upon approval of the duly completed Declaration of Conformity document and Supporting Evidence template.

2.4.4 Where necessary, CCC or the approved CLS(IoT) TL may choose to request for further clarifications or a presentation from the developer.

2.4.5 In the event of non-conformities, the developer may choose to resolve them, or the application shall be considered as unsuccessful for Level 1.

2.4.6 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 9.7 of CLS(IoT) Publication #1 – Overview of the Scheme [3].

3 LEVEL 2 – ADHERENCE TO INTERNATIONAL STANDARDS

3.1 OBJECTIVE

3.1.1 The objective of this activity is to ensure that the Device Under Test (DUT) conforms to a set of international standards.

3.1.2 **Level 2 is based solely on declaration of conformity by the developer**.

3.1.3 Devices that have completed Level 2 would entail that the developer has taken steps to ensure that the requirements within the specified international standard are met.

3.2 REQUIREMENTS

3.2.1 Level 2 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 The developer shall ensure that all mandatory requirements of the specified international standard are met.

3.3 DECLARATION OF CONFORMITY

3.3.1 Developers are required to complete and submit the following to declare conformity to the security requirements:

- Declaration of Conformity.
- Required supporting evidence.

3.3.2 The manufacturer shall refer to the “CLS(IoT) Publication #4 – Assessment Methodology” for details on the minimum requirement and the expected supporting evidence required for each provision. Some examples of supporting evidence include detailed descriptions, screenshots, process charts, work instructions. This expected supporting evidence are listed in the Supporting Evidence template.

3.3.3 The supporting evidence can be provided in the following forms:

- Provided in the entirety of the related documentation, with specific reference to the actual chapter/section/paragraph which contains the required supporting evidence to substantiate the claim to the meeting of the requirement.
- Provided in forms of screen captures or snippets of the actual document where the snippets shall contain the required supporting evidence to substantiate the claim to the meeting of the requirement.

3.4 ACCEPTANCE CRITERIA

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

3.4.2 However, CCC or the approved CLS(IoT) TL will review the submitted Declaration of Conformity document and Supporting Evidence template. Level 2 is only considered satisfied when CCC gains assurance through the submitted supporting evidence that the requirements are met.

3.4.3 The CLS(IoT) label is awarded by CCC upon approval of the duly completed Declaration of Conformity document and Supporting Evidence template.

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 all the non-conformities, or the application shall be considered as unsuccessful for Level 2.

3.4.6 Should any false declarations be subsequently discovered (possibly by the TL in subsequent testing or by other means), the testing laboratory shall inform the CCC, and CCC reserves the full rights to enforce actions as described in Chapter 9.7 of CLS(IoT) Publication #1 – Overview of the Scheme [3].

4 LEVEL 3 – LIFECYCLE REQUIREMENTS AND SOFTWARE BINARY ANALYSIS

4.1 OBJECTIVE

4.1.1 There are three components for this activity:

1. Meeting Mandatory Requirements. To ensure that devices meet the mandatory requirements of the specified international standard.

2. Lifecycle Requirements. To ensure that the developer adopts a “Security-by-Design” approach and implements adequate processes and practices to design, create, and maintain security on the Internet-Connected Device;

3. Software Binary Analysis. The test laboratory shall determine if the firmware and companion mobile application of the Device Under Test (DUT) is free from common software errors such as buffer overflown, known vulnerabilities in any of the third-party libraries being used, and known malware.

4.1.2 Devices that pass Level 3 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, and the device would likely be capable of resisting against script kiddies that leverages on readily available exploit kits.

4.2 REQUIREMENTS

4.2.1 Level 3 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 Developers are required to complete and submit the following to declare conformity to the security requirements:

- Declaration of Conformity.
- Required supporting evidence.

4.2.3 The developer is required to fulfil all 8 lifecycle provisions (CK-LP-01 to CK-LP-08) listed in the Declaration of Conformity.

4.2.4 The developer shall also provide the firmware and companion applications to the testing laboratory where they shall be subjected to testing under automated binary analysers which shall be performed by a testing laboratory.

4.3 PROCESS

Lifecycle Requirements

4.3.1 For **all** device categories, the developer shall complete and submit the Declaration of Conformity document to CCC or CLS(IoT) TL to declare conformity to lifecycle requirements.

4.3.2 The developer shall provide adequate supporting evidence in the Supporting Evidence template alongside the Declaration of Conformity document (e.g., detailed descriptions, screenshots, process charts, work instructions, etc.) such that CCC or CLS(IoT) TL is able to assess if the lifecycle requirements have been met, and that the security lifecycle processes and practices are adopted. Some examples of the expected supporting evidence are listed in the CLS(IoT) Assessment Methodology.

Software Binary Analysis

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

4.3.4 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.

4.3.5 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.).

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

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

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

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

4.4 SCOPE OF SOFTWARE BINARY ANALYSIS

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

Software Errors

4.4.2 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.

4.4.3 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.

4.4.4 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

4.4.5 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.

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

4.4.7 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.

4.4.8 It is expected that the developer shall provide the rationale and remediation taken to address each CVEs found. The rationale and remediation shall be provided to the test laboratory. Using the provided rationale and resolution, the test laboratory shall then perform the steps as mentioned in paragraphs 4.4.10 and 4.4.11.

4.4.9 The method of resolution could be any, but not limited to, the following:

- Perform a flaw remediation to address the discovered vulnerability. Examples of flaw remediation could be the patching of vulnerable components to address vulnerabilities, disabling vulnerable components, implementing technical measures to address vulnerabilities.
- If the discovered vulnerability is a false positive (e.g., the vulnerable component is not being used), the manufacturer shall provide this assessment to the laboratory. The test laboratory shall verify the suitability of this assessment and note it in the test report.
- Assess the vulnerability to be difficult/unexploitable. The assessment shall be provided to the test laboratory and the test laboratory will perform the first review of the suitability of this assessment.

4.4.10 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.).

4.4.11 The testing laboratory shall check that all components and respective vulnerabilities are accounted for. If the 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, where possible.

4.4.12 In the course of this procedure, the developer may choose to update libraries/components to a higher version, or may implement a custom patch/fix to address the vulnerability. In these situations, the testing laboratory shall perform a new scan of the binary code following developer’s remediation procedures. The remediated findings and its remediation steps must be included in the report to CCC.

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

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

4.4.15 The testing lab shall compare the software binary analyser results with the developers’ SBOM to identify and perform an analysis of the undetected components for vulnerabilities, ensuring that all vulnerabilities are accounted for and addressed by the developer.

Malware Scan

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

4.4.17 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.

4.4.18 If the 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

4.4.19 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.

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

4.4.21 The findings shall be resolved or justified as appropriately.

Search for Vulnerabilities in the Public Domain

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

4.4.23 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.

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

4.4.25 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.

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

4.5 ACCEPTANCE CRITERIA FOR LIFECYCLE REQUIREMENTS

4.5.1 CCC or CLS(IoT) TL will review the submitted Declaration of Conformity document and Supporting Evidence template. This procedure is only considered satisfied when CCC gains assurance through the submitted supporting evidence that the developer has implemented the required processes and practices and utilises them throughout the lifecycle of the DUT.

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

4.5.3 In the event of non-conformities, the developer may choose to resolve them, or the application shall be considered as unsuccessful for Level 3.

4.5.4 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 9.7 of CLS(IoT) Publication #1 – Overview of the Scheme [3].

4.6 PASS CRITERIA FOR SOFTWARE BINARY ANALYSIS

4.6.1 The firmware and the companion mobile application shall be free from identified exploitable vulnerabilities using the binary analysers. For non-conformity, 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.

4.7 TESTING LABORATORY DELIVERABLES

4.7.1 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

4.7.2 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.

4.7.3 During the course of testing, if the testing laboratory discovers any discrepancies or false declarations in the developer's declaration of conformity to the Security Baseline Requirements, International Standards, 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 9.7 of CLS(IoT) Publication #1 – Overview of the Scheme [3].

5 LEVEL 4 – BLACK BOX PENETRATION TESTING

5.1 OBJECTIVE

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

5.1.2 Devices that pass Level 4 should be capable of providing resistance against attacks conducted by a basic attacker on exposed interfaces.

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

5.1.4 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.

5.2 PRE-REQUISITES

5.2.1 Developers are required to complete and submit the following to declare conformity to the security requirements:

- Declaration of Conformity.
- Required supporting evidence.

5.2.2 The developer shall provide the following to the testing laboratory:

1. Firmware and companion applications
2. Guidance document (installation/operation guide).
3. Sufficient number of DUT to meet testing laboratory’s requirements.

5.2.3 The developer shall provide a single unit of the DUT to CCC. In the event of reports of security vulnerabilities for the DUT after the completion of the project, CCC may conduct internal investigations using the provided DUT.

5.3 SCOPE

5.3.1 This activity comprises the following tasks:

No.	Tasks
1	Device setup and verification of guidance documents
2	Conformity Verification – verifying that the device indeed implemented the security measures that the developer has declared and specified in the Declaration of Conformity document.
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 Level 3, b) Known threat vectors, c) The laboratory’s expertise and experience.
6	Password cracking (if applicable)

Table 2 – Level 4 tasks

5.3.2 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.

5.3.3 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.

5.3.4 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

5.3.5 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.

5.3.6 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.

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

Conformity Verification

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

5.3.9 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

5.3.10 To ensure consistent penetration testing of connected products across different testing laboratories, minimum test specifications for the different categories of connected products are defined.

5.3.11 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.

5.3.12 The testing laboratory shall take reference from CLS(IoT) Publication #5 – Minimum Test Specifications and Methodology for Level 4 [4] for this task. Supplementary Minimum Test Specification may be available for selected categories of products. Where such supplementary minimum test specification is available, the testing laboratory is required to include the additional tests.

5.3.13 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

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

5.3.15 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.

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

5.3.17 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

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

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

5.3.20 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 to make it malfunction and discover potential flaws.

5.3.21 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.

5.3.22 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.

5.3.23 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.

5.3.24 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

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

5.3.26 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

5.3.27 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 as an attempt to retrieve them.

5.4 PASS CRITERIA

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

5.5 DELIVERABLES

5.5.1 Prior to the beginning of any testing, the testing laboratory shall arrange a kick-off meeting with CCC to discuss the test approach and test plan.

5.5.2 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.

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

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

5.5.5 During the course of testing, if the testing laboratory discovers any discrepancies or false declarations in the developer's declaration of conformity to the Security Baseline Requirements, International Standards, 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 9.7 of CLS(IoT) Publication #1 – Overview of the Scheme [3].

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] Cyber Security Agency of Singapore, “CLS(IoT) Publication #1 - Overview of the Scheme,” CCC SP-151-1 Version 1.3, September 2023.

[4] Cyber Security Agency of Singapore, “CLS(IoT) Publication #5 - Minimum Test Specifications and Methodology for Level 4,” CCC SP-151-5 Version 1.1, April 2021.

ACRONYMS

The following acronyms are used in CLS(IoT) Publication 2:

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