CPNI
Centre for the Protection of National Infrastructure

CAPSS Application Notes for Manufacturers

PUBLISH DATE: June 2021

CLASSIFICATION: OFFICIAL

Cyber Assurance of Physical Security Systems (CAPSS) – 2021

Application Notes for Manufacturers

Updated v1.1

CPNI
Centre for the Protection of National Infrastructure

National Cyber Security Centre

Disclaimer

The information contained in this document is accurate as at the date it was created. It is intended as general guidance only and you should not rely on it. This information should be adapted for use in the specific circumstances required and you should seek specialist independent professional advice where appropriate before taking any action based on it. To the fullest extent permitted by law, CPNI accept no liability whatsoever for any loss or damage incurred or arising as a result of any error or omission in the guidance or arising from any person acting, relying upon or otherwise using the guidance. Full terms and conditions governing the use of this guidance are available on our website at www.cpni.gov.uk.

Freedom of Information Act (FOIA)

This document is authorised and issued by CPNI and NCSC

This information is supplied in confidence to the named reader and may not be disclosed further without prior approval from CPNI. This information is exempt from disclosure under the Freedom of Information Act 2000 (FOIA) and may be exempt under other UK information legislation.

Document history

CPNI may review, amend, update, replace or issue new CAPSS documents as may be required from time to time. There will be a regular review period to ensure that the requirements remain up-to-date. The CAPSS scheme is revised from time to time, creating new versions of each of the documents that define the scheme. Different versions of the CAPSS scheme as a whole are denoted by the year – hence for example “CAPSS 2019” or “CAPSS 2021” – version numbers of individual documents are issued within a version of the scheme hence both CAPSS 2019 and CAPSS 2021 have their own version 1.0 of the Application Notes.

Version	Date	Description
1.1	22 Jun 2021	Updated release
1.0	15 Jan 2021	First Release

Any comments or suggestions regarding this document should be directed to: cse@cpni.gov.uk

Contents

Executive Summary

This document provides advice and guidance on the application of the mitigations specified in the CPNI's 'Cyber Assurance of Physical Security Systems (CAPSS) 2021 – Security Characteristic'. It is intended for evaluation staff and assumes knowledge of the Security Characteristic (SC).

Section 1 – Overview

1.1 Introduction

This document provides guidance on applying the mitigations specified in CPNI's 'Cyber Assurance of Physical Security Systems (CAPSS) 2021 – Security Characteristic' when evaluating specific products. It is intended to help to ensure consistency of application of the mitigations across evaluations.

1.2 System description

The physical security systems covered by the SC are those that provide physical security measures while using IT systems and communicating over IP networks. These include Automatic Access Control Systems (AACS) Visitor Management Systems, Closed Circuit Television (CCTV), Intrusion Detection Systems, and Physical Security Information Management Systems. Each of these may employ distinct network services and protocols, distinct client and server elements, and a variety of sensors or other interface devices. Some elements will be deployed in a secure area while others will be deployed in public or non-secure areas. Some will be automatic while others will be attended or monitored by staff.

Therefore, as there is a wide variety of systems that are addressed by the SC, this document has been developed to provide guidance on applying the SC to the evaluation of specific products.

1.3 Expected operating environment

In most cases, a Physical Security System (PSS) will consist of a number of different products addressing various aspects of a protection objective, where each product may have been provided by one or more suppliers from one or more manufacturers. Figure 1 illustrates the types of elements that are likely to be included in such a system. Some elements will necessarily be deployed in exterior, public or otherwise non-secure areas, and will generally be unattended once deployed. Other elements such as controllers and management systems must be deployed in one or more secure areas. Some must be deployed in a secure enclave (such as a secured server room or a control room – see Appendix B Glossary). External services may be required, including provision of network connectivity, reliable time services, or for sending alarms to other organisations such as emergency services. Typically, subsets of products will be installed as a subsystem consisting of elements in both secure and non-secure areas, requiring communications between them. Such subsystems may operate independently or integrated with other subsystems.

Figure 2 shows an example implementation, where a command & control subsystem implements the integrated management, logging and admin functions; an AACS controller subsystem communicates with a variety of deployed interactive devices to permit access for authorised users; a CCTV recording subsystem receives feeds from CCTV cameras for monitoring; a physical Intrusion Detection System (IDS) deploys movement and infra-red sensors; a perimeter monitoring system (outside the secure enclave) deploys exterior sensors; and a Visitor Management System (VMS) subsystem manages access by visitors with a reception workstation.

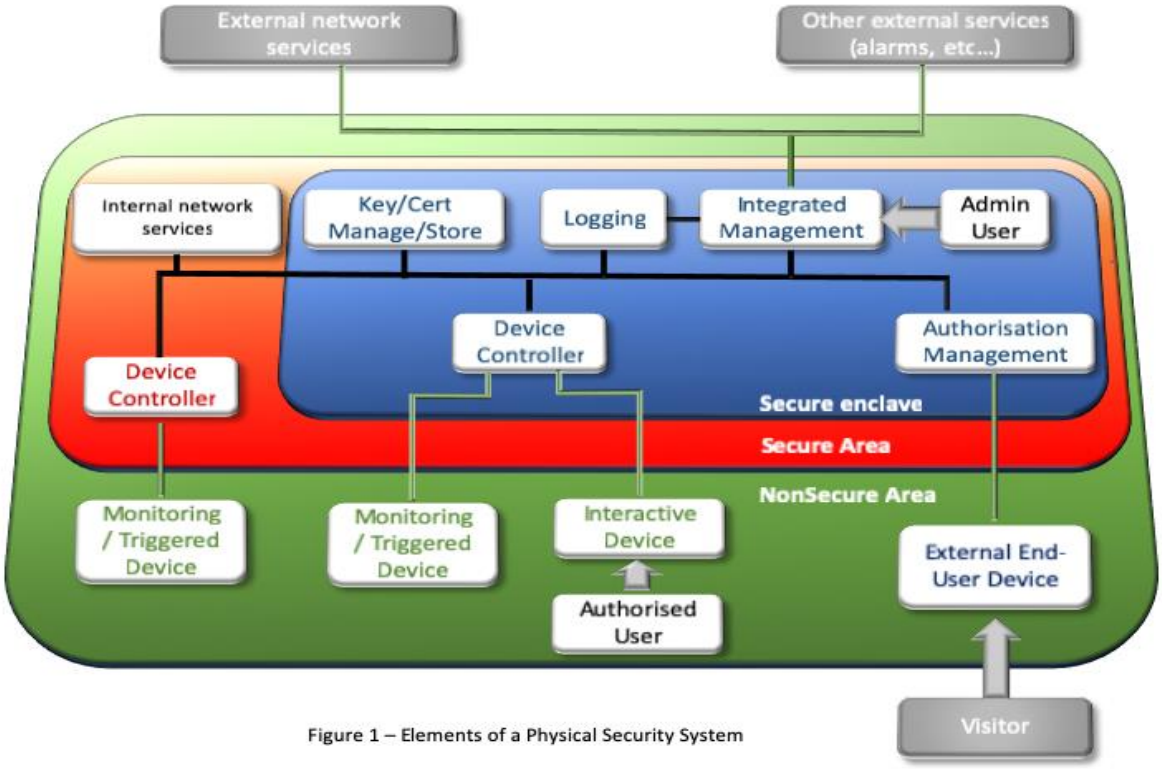

Figure 1 – Elements of a Physical Security System

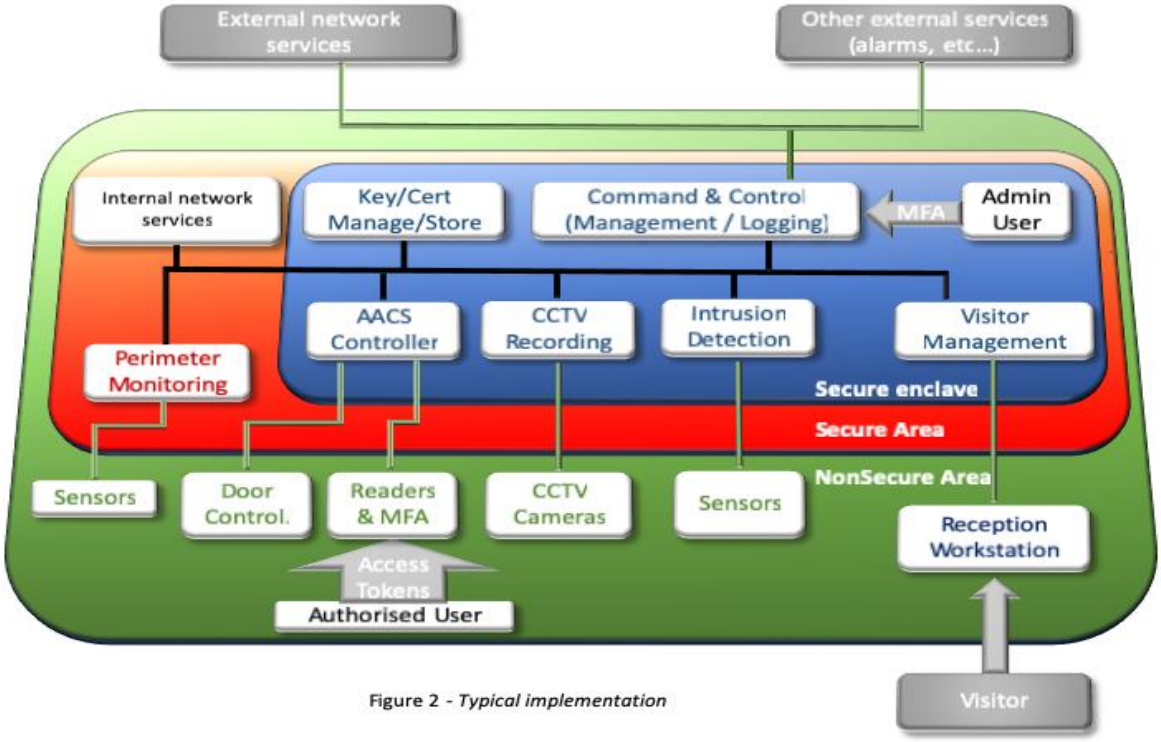

Figure 2 - *Typical implementation*

1.4 Secure enclaves and secure areas

While the distinctions between a non-secure area and secure area (see Appendix B Glossary) are self-evident, the distinctions between a secure area and secure enclave are limited to some specific aspects of particular requirements:

- **sensitive data should not be stored on devices that are exposed outside of the secure enclave [DEV.105/DEP.105];**
- **any communications link that is partially or entirely outside the secure enclave must be regarded as untrusted [DEV.406] and shall be included in the scope of fuzz testing [VER.407];**
- **admin access to subsystems that are deployed within the secure enclave, must also be within the secure enclave, while admin access to subsystems that are deployed outside the secure enclave but within a secure area, may be within the same secure area [DEP.204].**

1.5 How to use these Application Notes

Because the SC is based around a generic model and generic requirements, the evaluator first produces a Tailored Security Characteristic (TSC) that defines the requirements specific to the particular product being evaluated. These Application Notes provide guidance on applying the mitigations to specific products in order to produce the TSC. Because there is a significant benefit to potential end-users from understanding what mitigations have been applied, and to which elements of the product, it is expected that the TSC would be published as a separate document, and would **not** be part of an Assurance Plan. Note that this is therefore different from the 'normal' case described in NCSC's 'Process for Performing CPA Foundation Grade Evaluations (PPFGE)', section III E, (see Appendix B).

The evaluator will subsequently produce an Assurance Plan to identify how each of the mitigations in the TSC will be tested in the evaluation of the product. In general, compliance with Development mitigations are expected to be assessed from examination of the developer's design documentation, although other evidence may also be required; compliance with Verification mitigations is expected to be assessed by performing tests on the product or device; compliance with Deployment mitigations is expected to be assessed from installation and deployment guidance provided to end-users by the developer.

1.6 High level functional components and mappings

[SC, 1.8] identifies six functional components of a PSS and uses them to group the mitigations.

A bidirectional mapping between the SC functional requirements and the functional requirements in [62443-4-2], along with guidance for using IEC 62443-4-2 certification in CAPSS evaluations, is given in [Map, 3] ([Map] is available on request from CPNI).

1.7 Pre-requisites

[SC, 1.9] identifies the pre-requisites for a CAPSS evaluation (i.e. items that are required but that are not covered by the detailed DEV, VER, DEP and Build Standard requirements). The evaluators will assess and report on the way in which the developer satisfies each of the pre-requisites. Assessment of different aspects of the pre-requisites may be split between the Build Standard

Validation Report and the Evaluation Summary Report, but the Evaluation Summary Report shall state the final conclusion on achievement of the pre-requisites.

In general it is not necessary for the evaluators to report extensively on the pre-requisites when they are demonstrably met: it is sufficient to report an affirmation of the pre-condition and give a reference to the evidence provided to the evaluators.

Pre-requisite 2 in [SC, 1.9] relates to the presence of "a management system that encompasses information security" in the development environment, and identifies accreditation to [ISO27001] certification or Cyber Essentials PLUS [CEPlus] as suitable demonstrations of this (alongside ISO9001 accreditation). Other alternative accreditations for the cybersecurity aspects are acceptable in place of [ISO27001] or Cyber Essentials PLUS [CEPlus], provided that:

(i) they are publicly available standards;
(ii) their use in the organisation is demonstrated by independent audit and accreditation by a recognised accrediting body with a publicly available record of accreditation; and
(iii) the accredited scope includes the structures and processes that are involved in the production and delivery of the target PSS (and hence that this is within the scope of the independent audit).

The evaluators shall confirm and report the ways in which the criteria (i)-(iii) above are met by the developer when using alternative accreditations.

The subsections below identify some alternative standards that can be acceptable for CAPSS purposes, with particular topics that should be addressed for each in order to confirm their acceptance for a particular evaluation. They are therefore only directly relevant if those alternative standards are being used, but the subsections also provide examples of how other standards might be evaluated as acceptable equivalents.

It is not expected that alternative accreditations would give complete exemption from a Build Standard evaluation. Some procedures and practices may be common between other standards and CAPSS, and therefore evidence may be reused to support some aspects of the Build Standard where suitable.

1.7.1 NIST SP 800-171

NIST SP 800-171 is primarily concerned with protecting the *confidentiality* of information, whereas integrity and availability are also important issues for the development of a CAPSS target PSS. However, as stated in [SP800-171, footnote 19]: "The security objectives of confidentiality and integrity are closely related since many of the underlying security mechanisms at the system level support both objectives. Therefore, the basic and derived security requirements in this publication provide protection from unauthorized disclosure and unauthorized modification of CUI." This is combined with the presence of requirements covering integrity in [SP800-171, 3.14] on System and Information Integrity, and the fact that the Build Standard assessment (e.g. requirement 6) will separately assess other integrity aspects concerned directly with the target PSS. Availability of the product in the development environment is specifically addressed by Requirement 6.3 in section 4.2.3 below.

NIST SP 800-171 is based on the generic controls in NIST SP 800-53 ([SP800-53]), as applied to a minimum impact value of moderate (confidentiality). The acceptability of other uses of [SP800-53] for CAPSS systems will generally depend on how the generic criteria have been tailored using the

operations included in the template requirements and potentially other iteration and refinement actions (see [SP800-53, 2.2]), as well as the assumed minimum impact value.

An assessment methodology for SP 800-171 has been published by the US DoD at https://dodcmmc.org/. This methodology includes a 'Basic' level which is derived from self-assessment by the developer: this level would not meet requirement (ii) above for the acceptance of alternative accreditations for pre-requisite 2 in CAPSS.

The requirements of SP 800-171 may also be covered by certification under CMMC as described in section 1.7.3 below.

1.7.2 IEC 62443-4-1

This international standard is based on eight Practices related to design and implementation of Industrial Automation and Control System (IACS) products that need to provide defined security properties (usually those described in IEC 62443-4-2). As a development lifecycle standard, it has a stronger overlap with the Build Standard itself than with ISO 27001 or CyberEssentials Plus. However, the security culture on which it is based, and the practices concerned with security in the wider development environment (e.g. the requirements of Practice 1 (Security Management) that underlie the other practices) form a sound basis for addressing many of the other background controls.

Note that this standard does not address aspects such as classification and labelling of information (and related handling procedures). If these are particularly relevant to an evaluation it may therefore be necessary for the evaluators to evaluate these controls. IEC 62443-4-1 also does not address environmental aspects such as physical security, but these are covered in the CAPSS areas of interest by the Build Standard.

IEC 62443-4-1 is based not only on meeting the requirements in the Practices, but on a process maturity level (see [62443-4-1, clause 4.2]) which should be clearly identified in any accreditation to that standard. For the purposes of CAPSS evaluations this level should ideally be at least maturity level 3, although since the adoption of this standard is at a relatively early stage, level 2 may be more frequently found[1] and would be acceptable at the present time. Developers should expect that this requirement will be raised to maturity level 3 in future.

1.7.3 Cybersecurity Maturity Model Certification (CMMC)

The Cybersecurity Maturity Model used in [CMMC] has 5 levels that measure the maturity of cybersecurity processes and practices employed by an organisation across 17 'domains' (i.e. areas such as access control, physical protection, and system and communications protection). Each domain is associated with one or more 'capabilities' which, according to the CMMC level, leads to between 17 and 171 specific practices required of the organisation. Processes represent the intended ways that the organisation will operate to achieve defined objectives (e.g. as captured in

[1] In informal terms: level 2 represents an organisation that is implementing and following relatively new procedures, and it is therefore anticipated that there may turn out to be gaps in the application of the processes, or even in the definition of the processes themselves. Level 3 represents an organisation that has a strong demonstrable record of applying the procedures to a significant number of projects over a significant length of time – hence the processes are expected to be more complete and applied more consistently. Although the maturity level definitions only identify continuous improvement at levels 4 and 5, Practice 1 includes the SM-13 requirement for continuous improvement of the secure development process at all maturity levels, and hence this justifies an expectation that an organisation will increase its maturity level over time.

organisational procedures), while Practices are the activities actually carried out in order to achieve the defined objectives (and that therefore should leave auditable evidence of how the organisation *is* operating). CMMC is akin to NIST SP 800-171 in making confidentiality (rather than integrity) its primary concern, and [CMMC, 2.7.1] states that all the requirements of SP 800-171 are covered at CMMC level 3 and above. CMMC levels 4 and 5 go further by taking steps to reduce the risk from APTs.

Levels above CMMC level 1 require that the organisation has processes in place to document, plan, and reflect on their requirements and actions in each domain. The processes move from simple documentation, to optimisation and common implementation across the organisation as the CMMC level rises.

Because CMMC is publicly available (including Assessment Guides with specific assessment objectives for each individual practice) and is based on independent audit and accreditation of the maturity level[2], it meets requirements (i) and (ii) for acceptability of alternative accreditations for pre-requisite 2, as discussed in section 1.7 above.

A CMMC certification at level 3 is therefore potentially acceptable as an alternative accreditation for pre-requisite 2. However, the suitability of the scope for requirement (iii) must be confirmed within a CAPSS evaluation.

1.8 Security Procedures for CAPSS

For CAPSS evaluations (unlike CPA evaluations as described in [PPFGE]) it is generally expected that the developer will produce the Security Procedures documentation necessary to meet the DEP requirements. This does not preclude the evaluators from producing them, or from updating them as a result of evaluation activities. The Security Procedures may take the form of existing product documentation provided that it meets the CAPSS requirements, describes the CAPSS required configuration, and is maintained under assurance maintenance for delivery to customers. In particular, parts of the documentation that have been used to meet DEP requirements must only be updated under an assurance maintenance process where the impact of changes is independently impact-assessed.

1.9 Additional information

This document has been produced by CPNI with input from, and review by, NCSC.

1.10 Information on future changes

Where accreditation to IEC 62443-4-1 is used as a pre-requisite, as described in section 1.7.2, it is expected that in future the required process maturity level will be a minimum of level 3.

With regard to DEV.108 (Protected software environment): requirements on 3rd party products are expected to be strengthened in future. It is intended that 3rd party software (including firmware) used in the target product will also be subject to MISRA analysis, either by the target product developer or the 3rd party developer. Some of the issues related to 3rd party software are discussed in [MISRA Comp, 6] on “Adopted Code”.

[2] The CMMC Accreditation Body has a web site at https://cmmcab.org/

With regard to DEV.402: [IEEE802.1X] may be required over MAC filtering in future.

With regard to VER.407: it is preferred that developers undertake their own fuzz testing as part of one or more stages in the product development lifecycle. Fuzz testing by the developer may become a mandatory requirement in future.

Section 2 – About Application Notes

2.1 Application Note format

This CPNI Application Notes document includes guidance for interpretation of the SC requirements, in particular:

- **suggested criteria to determine the applicability of the mitigations defined in the SC to a specific product. These criteria are summarised in section 2.2 and should be used in the production of a TSC for the product to be evaluated**
- **constraints expected on the implementation of the mitigations, in order to achieve the objective of the mitigation in practice**
- **details of the evidence expected from the developer (and sometimes from the evaluator) in order to demonstrate that a mitigation has been correctly implemented.**

The mitigations are presented in the same order as in [SC, 3] in three requirement categories (development, verification and deployment) and further grouped into the functional areas to which they relate (as described in [SC, 1.8]). Note that where more than one item of evidence is listed it is intended that all identified items are required as evidence, unless they are specifically identified as alternatives.

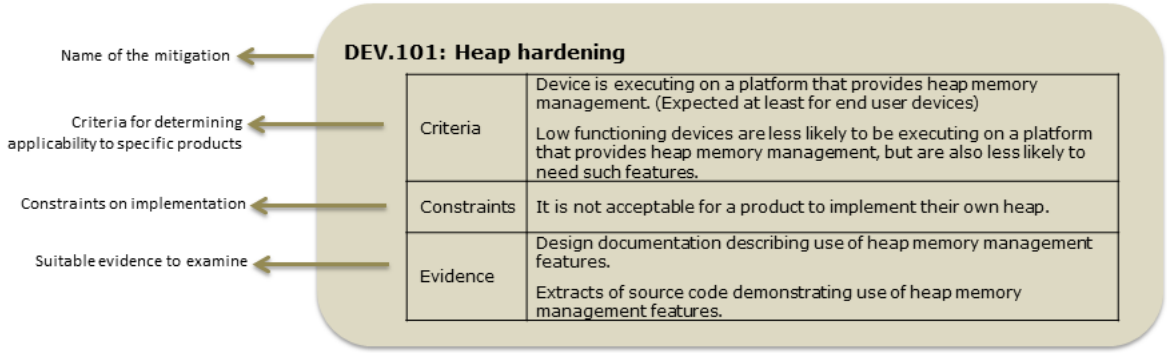

Figure 1 – Components of a typical entry

An asterisk against the identifier in the mitigation name indicates that it is included in the list of mitigations in [SC, Appendix D] that allow a rationale for non-implementation of the mitigation.

2.2 Determining applicability of a mitigation in practice

In practice, determining the applicability of each mitigation in the SC to a particular part[3] of a PSS under evaluation (the "target part" of the "target PSS") is based on a 3-step analysis:

[3] Note that the term "component" is avoided for describing the parts of a PSS because it is used in the language of CPA to define functional component groupings of mitigations – see [SC, 1,8]

- Identify whether the target part uses cloud services: this determines whether or not DEV.700, VER.700 and DEP.700 apply to the target part
- Identify (based on the description in [SC, 1.5]) the intended deployment location of the target part in terms of the expected operating environment, i.e. one of secure enclave, secure area or non-secure area. The outermost (i.e. least protected) of the environments in which the component is intended to be used should be used for this identification.
- Identify, for each target part, any mitigations that are not met due to the level of functionality of the device (i.e. where it is a low functioning device as described below) and/or because the threat does not apply to its intended deployment location (in this case a rationale is required)

  This will include using the guidance in [SC, Appendix D] and consideration of whether the target part is high functioning or low functioning according to the following criteria:

  - High functioning – a device or product executing in an environment with operating system and/or hardware support for security enforcing functionality.
    - Typical examples would be a PC running Windows or OSX, a server running Unix, or a tablet or smartphone running Android or iOS.
  - Low functioning – a device or product executing in a highly constrained environment, with no (or minimal) operating system support.
    - Typical examples would be sensor devices implemented as a simple circuit, an FPGA/ASIC device or a simple device with minimal firmware.

  End user devices and servers would therefore always be considered as high functioning, but other parts of a target PSS might depend on the presence or absence of certain features, such as the examples of *DEV.102: Stack Protection* and *DEV:105 Encrypt sensitive data* discussed in [SC, Appendix D]. The mitigations listed in [SC, Appendix D] as allowing a rationale for non-implementation of the mitigation are marked with an asterisk against the mitigation identifier in the tables in section 3 of these Application Notes (e.g. “DEV.101*: Heap hardening”). The Criteria rows in the tables below are also intended to help in the conclusion as to when a mitigation applies, and include a note in the form “(Expected at least for end user devices)” for requirements that would generally be expected to apply to an end user device – the requirement could of course also apply to other types of device.

  As a further example (independent of the high/low functioning distinction), *DEV.406: Encrypt communications traffic over untrusted link* might not apply in some cases where either no sensitive data is transmitted or the link is sufficiently protected by other means. But if the device or its functionality is susceptible to the threat of ‘interception of sensitive data from unencrypted links’ then it must use encryption on those links.

Section 3 – Evaluation Activities

3.1 Development mitigations

3.1.1 Development >> General

DEV.100: Evaluation/Cryptocheck

Criteria	This is applicable to all devices.
Constraints	Cryptographic Algorithm Validation Program (CAVP) Validation of all cryptographic algorithms used for security functionality in the product is required before the evaluation commences. This must include all cryptographic algorithms used in communications protocols. If the cryptographic algorithms in use in the product have not been certified under CAVP, or equivalent external certification, the developer must discuss suitability with CPNI before the start of the evaluation. CPNI will confirm suitability of the implementation with NCSC before the evaluation can proceed.
Evidence	Rationale for the choice of cryptographic algorithms used in the product, and evidence that they have been independently validated for correctness under CAVP (or equivalent external certification).

DEV.101*: Heap hardening

Criteria	Device is executing on a platform that provides heap memory management. (Expected at least for end user devices) *Low functioning* devices are less likely to be executing on a platform that provides heap memory management, but are also less likely to need such features.
Constraints	It is not acceptable for a product to implement their own heap.
Evidence	Design documentation describing use of heap memory management features. Extracts of source code demonstrating use of heap memory management features.

DEV.102*: Stack protection

Criteria	Device uses stacks. (Expected at least for end user devices) *Low functioning* devices are less likely to be implemented using a tool chain that supports stack protection. In such devices it is possible that stacks are not used at all. The level of functionality of the available device interfaces (whether or not the interfaces are actually used in the target system) – such as the types of objects, parameters, and message content constraints – may be taken into account when judging the relevance of the requirement.
Constraints	If stacks are in use on the device and the tool chain does not support stack protection, the equivalent measures must be implemented by the developer.
Evidence	Design documentation describing use of tool chain stack protection features, or equivalent features implemented by the developer. Developer's coding guidelines identifying any action required to use stack protection. Build logs confirming correct settings are applied during build.

DEV.103*: Data Execution Prevention

Criteria	Device is executing on a platform that supports either Software Data Execution Prevention or Hardware-enforced Data Execution Prevention, or equivalent (such as W^X). (Expected at least for end user devices) *Low functioning* devices are less likely to be executing on a platform that supports Data Execution Prevention or an equivalent. The level of functionality of the available device interfaces (whether or not the interfaces are actually used in the target system) – such as the types of objects, parameters, and message content constraints – may be taken into account when judging the relevance of the requirement.
Constraints	If the underlying platform supports Data Execution Prevention or an equivalent, the product must not opt out.
Evidence	Design documentation describing use of Data Execution Prevention or equivalent. Developer's coding guidelines identifying the use of Data Execution Prevention and how to avoid undermining its effectiveness. Build logs confirming correct settings are applied during build.

DEV.104*: Address Space Layout Randomisation

Criteria	Device is executing on an operating system that supports Address Space Layout Randomisation (ASLR). (Expected at least for end user devices) *Low functioning* devices are less likely to be executing on a platform that supports ASLR. The level of functionality of the available device interfaces (whether or not the interfaces are actually used in the target system) – such as the types of objects, parameters, and message content constraints – may be taken into account when judging the relevance of the requirement.
Constraints	If ASLR is supported by the underlying operating system it must be used throughout the product, including in all libraries.
Evidence	Design documentation describing features used to implement ASLR. Justification and rationale if ASLR is disabled for any component of the product. Developer's coding guidelines identifying any action required to use ASLR. Build logs confirming correct settings are applied during build.

DEV.105*: Encrypt sensitive data

Criteria	If sensitive data is stored on the device then this requirement applies, including to devices in a non-secure area. Sensitive data is defined in the Glossary. (Expected at least for end user devices)
Constraints	Sensitive data should not be stored on devices that are exposed outside of the secure enclave. Sensitive data must be encrypted using hardware-backed encryption where available (e.g. TPM or TEE), otherwise using software encryption. The evaluators shall check the product's list of sensitive data types for identifiable omissions based on other design information, user guidance documentation, and on their own experience of using the product for VER tests.
Evidence	For cases where sensitive data is stored on a device that is, or may be, used outside a secure area or secure enclave, a justification for the need to store the sensitive data. Design documentation describing the types of sensitive data stored and/or processed by the PSS, the measures taken to protect each type, and the technical mechanisms used by functions of the host platform used to protect sensitive data (e.g. encryption algorithms, key management and generation of algorithm parameters such as initialisation vectors and cryptographic nonces). The sensitive data list is shared with other mitigations such as DEV.203 (Protection of security-related physical structure), DEV.406 (Encrypt communications traffic over untrusted link), DEV.600 (Log all relevant events) and DEP.105 (Encrypt sensitive data).

DEV.106: Updateable product

Criteria	In exceptional cases, some *Low functioning* devices could be incapable of supporting software updates. A consideration in assessing whether an update capability is required will be the likelihood of design errors being identified after deployment, and hence the complexity of the product may be taken into consideration. Support for updates includes consideration of any debug or similar interfaces present in the product.
Constraints	A software update mechanism must be implemented unless it is infeasible, in which case justification must be provided.
Evidence	Design documentation describing the software update mechanism. Design documentation describing the presence and state of any debug or similar interfaces in the product (cf. DEV.200). Justification and rationale in the case of *Low functioning* devices that are incapable of supporting software updates.

DEV.107: Secure software delivery

Criteria	In exceptional cases, some *Low functioning* devices could be supplied with pre-installed software and be incapable of supporting software updates.
Constraints	A cryptographically protected software delivery mechanism must be implemented unless the device is supplied with pre-installed software and software updates are not possible, in which case justification must be provided. The software must be signed in such a way that it can be verified before installation or applying an update.
Evidence	Design documentation describing the software delivery mechanism. Justification and rationale in the case of *Low functioning* devices that are incapable of supporting secure software delivery.

DEV.108*: Protected software environment

Criteria	Device is executing on a platform or operating system that provides defensive or robustness mechanisms. (Expected at least for end user devices) *Low functioning* devices are less likely to be executing on a platform or operating system that provides defensive or robustness mechanisms. In this case software/firmware security review is still expected.

Constraints	Defensive or robustness mechanism provided by the platform or operating system are expected to be employed by the product. Conformance to MISRA rules should be analysed by an automated tool. It is noted that conformance to the MISRA standard includes the content in [MISRA, 5.5 & 6.2] and updated in [MISRA Comp][4] which explains what it means to claim conformance, describes the different categories of guidelines and stipulates, for example, that "deviation from mandatory guidelines is not permitted". Where deviations from other guideline categories are present (and where any guidelines are 'disapplied' [5]) the developer shall provide a rationale describing (a) why the deviations do not represent a significant risk to the PSS security in practice, and (b) why it is not practical to modify the code to conform to the guideline. In assessing the rationale, as well as assessing whether the argument for (a) and (b) is convincing, and considering the discussion of deviations in [MISRA Comp, 4], the evaluators should also ensure that they are convinced that the rationale applies given the *number* of deviation instances. For example, the number of instances might affect the practicality of inspecting and confirming each instance by eye; and a developer's rationale should not be based simply (or primarily) on the time taken to update a large number of instances. Where a strong case can be made by the developer, it may be appropriate (at the discretion of CPNI) to take into account a history of continuous improvement in reducing MISRA deviations in the product combined with a specific plan for future reductions. In some cases deviations might be identified as false positives (e.g. where the analysis tool cannot properly determine the context of a piece of code in which an apparent deviation arises). Such cases also require a rationale (at least for mandatory and required categories). The investigation of any rationale is likely to require sight of a selection of coding samples of the evaluators' choosing to determine whether there are more systemic issues with the developers' coding practice that may be resulting in anomalous amounts of deviation. The same principles (including the principles from [MISRA Comp]) can be applied to evaluation of alternative, equivalent approaches to MISRA (e.g. where C is not the language used in the product).
Evidence	Design documentation describing process environment and defensive and robustness mechanisms provided by the platform and operating system. Evidence of software / firmware review against known vulnerabilities. Evidence from static tool analysis / lint-like tool for device software and firmware and third party libraries / components. Results of Build Standard Validation (requirements 11, 13).

[4] Note that [MISRA Comp, 1] states that it "supersedes the compliance, deviation and process requirements published previously in the various MISRA Guidelines."

[5] Cf. [MISRA Comp, 5.1 & 6.4]

DEV.109: Unique security data per device

Criteria	Device contains keys or other security credentials.
Constraints	Keys or security credentials must not be shared across multiple devices.
Evidence	Design documentation describing the purpose, generation, installation and storage of keys and other security credentials. Evaluators should check that each identified key is generated, stored, and ultimately destroyed in a suitable way. Manufacturing processes and procedures may be relevant if keys and/or credentials are installed in the device at manufacturing time. This needs to ensure good generation processes, secure storage of any retained or pre-generated data, and secure handling of access to any relevant artefacts (e.g. key files or smart cards holding master keys) in the manufacturing environment.

3.1.2 Development >> Physical Security

DEV.200: Disable non-operational logical and physical interfaces

Criteria	This is applicable to all devices.
Constraints	Interfaces other than those required for normal operation must be either disabled or unable to be used to undermine device security. Debug interfaces must be disabled and any re-enablement must require either multi-factor authentication or else a visible change to the device, and an alert (this is similar to the tamper response requirement in DEV.201, and it would typically be expected that debug interfaces are inside the tamper boundary). The visible change might be achieved by, for example, seals, or snap joints, or the removal of an epoxy coating. The point of the visible change is to minimise the chance that the re-enabled state might be forgotten or remain unnoticed. Devices (including *Low functioning* devices) must not allow debug re-enablement by other means such as a simple jumper (although this is visible, it is unlikely to be externally visible and unlikely to involve sufficiently different device appearance to achieve the objective of minimising the chance that the re-enabled state might be forgotten or remain unnoticed). Hence re-enablement of debug interfaces on such simple devices that cannot support the requirement for physical modification followed by authentication must not be possible by any other means. In devices intended to be deployed in a non-secure area, disablement may be achieved by the use of epoxy potting over the debug interfaces to prevent their use. In devices intended for deployment in a secure area, that could not be deployed in a non-secure area, robust tamper detection may be considered a sufficient alternative.
Evidence	Design documentation describing all roles and interfaces used during any stage of the product lifecycle. This must include identifying messages on normal interfaces that put the device into an alternate mode (such as configuration mode or programming mode). Design documentation describing disablement of interfaces not required for normal operation, including alternate modes, debug interfaces and physical media. The means of disablement must be confirmed. This should include, for example, USB interfaces which should be disabled by default for devices in the secure area and secure enclave. All identified disabled interfaces must be included in testing for VER.200. Justification and rationale of assurance that non-disabled interfaces cannot undermine device security. Any such interfaces must be identified for inclusion in fuzz testing for VER.407. *Evaluators may consider it necessary to ask for additional support from the developer (e.g. tools or witnessing a developer test) to confirm that an interface has actually been disabled.*

DEV.201: Tamper response

Criteria	Tamper response is included for devices located in a secure enclave as a layer of protection against potential insider threats. Subject to agreement with CPNI, environmental mitigations that can reliably be assumed to exist within a secure enclave for the target device may be taken into account to mitigate some threats; but this is on a case-by-case basis. End-user devices that are protected by appropriate measures specified in [EUD] guidance (and servers or other high functioning devices implementing equivalent measures) to encrypt local data, such as Bitlocker, are not required to generate a tamper alert but their disconnection from a controller must be alerted by the controller.
Constraints	Attempts at tampering must be alerted and logged.
Evidence	Design documentation identifying the tamper-protection boundary and any part that is designed to be opened or removed. Design documentation for end-user devices confirming the use of appropriate measures as specified in [EUD], and alerting by controller on disconnection. Other high functioning devices should be able to justify their measures by reference to [EUD], with particular reference to the security principles (Data-in-transit protection, Data-at-rest protection, Authentication, Secure boot, Platform integrity and application sandboxing, Application allow listing, Malicious code detection and prevention, Security policy enforcement, External interface protection, Device update policy, Event collection for enterprise analysis, Incident response).

DEV.202: Fail secure on power loss

Criteria	This is applicable to all devices.
Constraints	In the event of a loss of power the device must not fail in a way that undermines the security requirements. When power is restored, the device must restart in a state that does not undermine the security requirements.
Evidence	Design documentation describing behaviour on power loss and power on.

DEV.203: Protection of security-related physical structure

Criteria	End-user devices that are protected by appropriate measures specified in [EUD] guidance (and servers or other high functioning devices implementing equivalent measures) to encrypt local data, such as Bitlocker, are not required to have a tamper-protection boundary.
Constraints	All components that generate, process or store sensitive data, or carry out cryptographic operations, must be inside the tamper-protection boundary. (The list of sensitive data handled by the PSS is required as part of DEV.105 (Encrypt sensitive data).)
Evidence	Design documentation identifying the tamper-protection boundary and methods and mechanisms used to provide protection. Design documentation identifying all cryptographic keys and their storage locations, confirming that all components that generate, process or store sensitive data, or carry out cryptographic operations, are inside the tamper-protection boundary. Design documentation identifying all physical and logical interfaces and input or output paths that are available across the tamper-protection boundary. Design documentation for end-user devices confirming the use of appropriate measures as specified in [EUD]. Other high functioning devices should be able to justify their measures by reference to [EUD], with particular reference to the security principles (Data-in-transit protection, Data-at-rest protection, Authentication, Secure boot, Platform integrity and application sandboxing, Application allow listing, Malicious code detection and prevention, Security policy enforcement, External interface protection, Device update policy, Event collection for enterprise analysis, Incident response).

3.1.3 Development >> Secure Configuration

DEV.300: Provide a configuration tool to enforce required settings

Criteria	This is applicable to all devices.
Constraints	If options requiring to be set by administrator exceeds 12, then the developer must supply a tool, policy template, or specific configuration guide which helps the administrator to achieve this in fewer steps. If the configuration tool can be used to change the product configuration after installation in the deployment environment then it must use only an authenticated interface (e.g. DEV.504 & DEV.505) to the product, to ensure that only authorised users can make such changes.
Evidence	Design documentation identifying the necessary configuration items and their values to cover all relevant requirements in DEP mitigations, and describing any applicable authentication.

DEV.301: Ensure product security configuration can only be altered by an authenticated system administrator

Criteria	This is applicable to all devices.
Constraints	Must ensure that only authenticated administrator can change the product's security settings. This includes configuration of any key and certificate management required in support of authentication or other cryptographic functionality.
Evidence	Design documentation and administrator guidance identifying the paths through which security enforcing configuration settings can be altered, and the authentication required to access them. Design documentation confirming the protection applied to configuration settings to protect against file modification or registry changes.

DEV.302: Ensure product security configuration can be backed up

Criteria	This is applicable to all devices.
Constraints	Restoration of a backup must be restricted to an authorised and authenticated security administrator. Backups of security configuration must be protected against modification by unauthorised users. In exceptional cases, some *Low functioning* devices could be incapable of supporting backup of security configuration. A consideration in assessing whether a backup capability is required will be the likelihood of default settings (e.g. after a reset) undermining security requirements, and hence the complexity of the product may need to be considered when judging the relevance of the requirement.
Evidence	Design documentation and administrator guidance confirming that security configuration can be securely backed up and restored in a timely fashion.

DEV.303*: Deploy onto suitably protected endpoint

Criteria	This is applicable to all endpoint devices. (Expected at least for end user devices)
Constraints	If the endpoint device is provided with the product, the developer must provide assurances that the relevant NCSC [EUD] Guidance for the platform has been met or, if such guidance is not available, then provide a rationale that they implement best practice for the platform.
Evidence	Assurance that the NCSC [EUD] Guidance for the platform has been met, or rationale that best practice for the platform has been implemented.

3.1.4 Development >> Network Security

DEV.400: Minimise interfaces

Criteria	This is applicable to all devices.
Constraints	Network ports and services must only be opened if required for the device to function.
Evidence	Design documentation identifying protocols and services available on all interfaces and their functional purpose. Justification and rationale of assurance that any other interfaces cannot undermine device security.

DEV.401: Wireless network must be secured

Criteria	This is applicable to all devices.
Constraints	WiFi connections must use WPA2 Enterprise as a minimum. Other wireless networking protocols must enforce use of secure protocols such as TLS employing NIST approved cryptographic algorithms. The requirement not to use wireless technologies on any site requiring more than a basic level of protection (cf. DEP.408), means that if it is necessary to disable any wireless capabilities in the product to meet the Tailored Security Characteristic then it may be relevant to check that the disabling mechanism is described in the design (cf. DEP.401).
Evidence	Design documentation identifying any wireless technology in use, and the measures used to secure wireless communications.

DEV.402: Use whitelist to limit communications

Criteria	This is applicable to all devices.
Constraints	The device should use a whitelist feature to ensure that communications are from authorised devices. Although MAC filtering is acceptable, [IEEE802.1X] is preferred and may be mandated in future versions of the CAPSS standard.
Evidence	Design documentation identifying the authorisation and authentication mechanism for establishing communications between devices.

DEV.403*: Use time synchronisation

Criteria	(Expected at least for end user devices) *Low functioning* devices are less likely to be executing on a platform or operating system that provides time synchronisation mechanisms. In this case a suitable mechanism should be implemented by the developers where possible.
Constraints	The time synchronisation can either be obtained from an external time server or from an internal time server with a trusted time source, using a suitable protocol such as NTP or PTP. The protocol in use must be a major version that is still supported, for which all up to date security patches have been applied.
Evidence	Design documentation identifying the mechanism employed and means of ensuring that the time source is reliable and trusted – this should include the rationale (such as identifying authenticated time sources or comparing multiple independent sources) for protecting against time spoofing attacks.

DEV.404*: Use segregated networks

Criteria	(Expected at least for end user devices) Many devices are unlikely to support multiple physical network connections, but can support logical segregation. *Low functioning* devices with a single interface may be unable to support multiple network connections. The level of functionality of the available device interfaces, and the lack of a management interface may be taken into account when judging the relevance of the requirement.
Constraints	If the product is supplied with network setup, this must use VLANs or other network segregation approaches to separate unrelated components. Any management interface must be on a separate VLAN.
Evidence	Design documentation identifying the required network setup and the mechanism(s) employed for separating unrelated components.

DEV.405*: General resource management

Criteria	(Expected at least for end user devices) *Low functioning* devices may be executing on a platform or operating system that provides little control over interrupt handling or management of input buffers. In this case the application is expected to manage the incoming traffic in any way possible (including temporarily ignoring input).
Constraints	The loss of external communications is more acceptable than loss of functionality resulting from a crash or general failure due to large amounts of incoming network traffic.
Evidence	Design documentation identifying the mechanisms in place to handle incoming network traffic, including a rationale identifying the impact of excessive traffic.

DEV.406*: Encrypt communications traffic over untrusted link

Criteria	(Expected at least for end user devices) Any communications link that is partially or entirely outside the secure enclave must be regarded as untrusted. In exceptional circumstances, some *Low functioning* devices may be executing on a platform that cannot support strong cryptography. The level of functionality of the device interface and the message content should be taken into account when judging the relevance of the requirement.
Constraints	Non-sensitive data needs to be provided with integrity protection at minimum. Sensitive data must be encrypted and integrity protected. (The list of sensitive data handled by the PSS is required as part of DEV.105 (Encrypt sensitive data).) The cryptographic algorithms and cipher suites used must be NIST approved.
Evidence	Design documentation identifying the protection applied to communications traffic over any link, including configurable options. Evidence that cryptographic algorithms in use are NIST approved. Installation, deployment or administration documentation identifying any configuration or administrative action required to configure the device to use the appropriate level of protection.

3.1.5 Development >> Authentication Management (Privileges)

DEV.500*: Role based access control

Criteria	(Expected at least for end user devices)
Constraints	Users must be able to be assigned to specific roles, with the roles determining what operations may be performed, ensuring that users are only able to perform operations and access data appropriate to their role. If the definition of user roles is customisable, this must only be able to be performed by an admin user with an appropriate privilege.
Evidence	Design documentation identifying the user roles provided and the functions and privileges available for each role. In general it is expected that there will be a minimum of two roles, an administrative role and a standard user role, although some *Low functioning* devices may require no more than an administrative role to configure the device and no subsequent user interaction.

DEV.501*: User least privilege

Criteria	(Expected at least for end user devices)
Constraints	For a non-administrative role, the product must operate correctly from a standard account without elevated privileges. Privileges include both OS and product-defined privileges.
Evidence	Design documentation identifying and justifying any elevated privileges that are required for specific functions or specific user roles (including administrative).

DEV.502*: User authentication

Criteria	(Expected at least for end user devices)
Constraints	If users are not required to use a MFA authentication mechanism that is unique to each user, there must be a password policy that, as a minimum, meets the requirements defined in Appendix C of the [SC]. The product must lock out a session after a defined period of inactivity, requiring the user to re-authenticate. Inactivity period may be configurable but must be no longer than 15 minutes for admin roles and any roles used outside the secure area; but may be up to 120 minutes for roles that are used in a secure area for passive review of data (such as CCTV).
Evidence	Design documentation identifying the user authentication mechanisms available. If the options available include a MFA mechanism, it must use factors that are unique to each user. If the options available include a password policy, it must be capable of being configured by the system administrator to be at least as strong as that defined in the [SC]. Options must include the ability to lock an account after a defined number of consecutive invalid login attempts. Design documentation confirming that a user session is locked after a defined period of inactivity, requiring the user to re-authenticate, including identifying the length of the inactivity period(s) and, if configurable, what limits are imposed. Design documentation identifying all passwords for which default values are defined in the product; and confirming that there are no other features available that use accounts with fixed passwords or equivalent fixed access credentials; and confirming that there is no fixed authentication data (such as 'magic numbers') in use in the product.

DEV.504*: Local management authentication

Criteria	(Expected at least for end user devices)
Constraints	Administrator accounts must use MFA authentication that is unique to each user. It is accepted that use of multi-factor authentication may be difficult to achieve at the current time for some low-functioning devices (this may include devices such as PIR sensors provided that they are low-functioning). At the discretion of CPNI, at the current time such devices may be accepted on the basis of implementing robust authentication based on whitelisting or the use of pre-loaded certificates, provided that they are unique to each user. ("Unique to each user" here means that (i) the same credentials must not be applicable to devices owned and/or managed by different entities (e.g. authentication details issued to company A must be different from those issued to company B), and (ii) that there must be separate administrator accounts for each administrative user unless an acceptable rationale is provided that a single administrator account is sufficient for local management of the device in practice.) If the product uses a separate management component (for monitoring and/or managing configuration and/or operation), then this must use an authenticated interface such as in DEV.504/DEV.505.
Evidence	Design documentation identifying the user authentication mechanism for administrator users and confirming that MFA authentication is available for use for administrative users.

DEV.505*: Remote management authentication

Criteria	(Expected at least for end user devices)
Constraints	Remote management access must be protected by a secure protocol and MFA authentication. This should follow (and/or should enable the deployment to follow) NCSC guidance such as [RM_NCSC], [IPsec_NCSC] and [TLS NCSC]. The version(s) of the protocol supported should be recent versions and should exclude any earlier versions known to be vulnerable. If the product uses a separate management component (for monitoring and/or managing configuration and/or operation), then this must use an authenticated interface such as in DEV.504/DEV.505.
Evidence	Design documentation identifying the remote management interface, the protocol(s) implemented to protect access and the authentication mechanism in place.

3.1.6 Development >> Monitoring

DEV.600*: Log all relevant events

Criteria	This is applicable to all devices.
Constraints	Logs here are intended to cover event and information logs rather than diagnostic or debug logs. Log data must be detailed enough to allow forensic investigation during any incident management. Sensitive data such as passwords and keys must not be written to the logs (the list of sensitive data handled by the PSS is required as part of DEV.105 (Encrypt sensitive data).). The [SC] identifies a list of the events to be logged as a minimum. But in producing a Tailored *Security Characteristic for a specific product evaluation, the evaluators shall determine the specific events of interest for each element.*
Evidence	Design documentation identifying the logging mechanism, and any configuration controls over the content of log entries and the events that are logged.

DEV.601*: Protect access to logs

Criteria	This is applicable to all devices.
Constraints	All log entries must be time stamped with time that is accurate and synchronised with a reliable time source. Only an authenticated administrator should be able to read log entries. It must not be possible to delete entries. The administrator must be alerted before logs are overwritten. Some simple devices with memory constraints may treat the log as circular, causing older entries to be overwritten by the latest entry if the log is full; in this case the log must be capable of holding at least 100 entries and must be exported to another device (such as a controller or central logging facility) regularly enough that log entries are unlikely to be lost. The overwriting of log entries in this way is acceptable provided that the developer supplies a valid justification for this behaviour, the size of the log and the frequency of export.
Evidence	Design documentation identifying the time source used for time stamping the log entries, including any configuration options available. Design documentation identifying the access controls applied to logs, demonstrating that only authenticated administrators can read logs or manage their export or backup, and identifying the measures implemented to ensure that log entries cannot be modified or deleted. Design documentation identifying the behaviour when logs are to be overwritten, demonstrating that the administrator is alerted and provided with the opportunity to ensure that the log files have been backed up or exported before they are overwritten. For a *Low functioning* device where the log is implemented as a circular buffer, design documentation providing details and justification of the size of the log, the behaviour when full and the export mechanism.

DEV.602*: Export logs

Criteria	This is applicable to all devices.
Constraints	The product must provide the ability to automatically transfer log records to an external device and protect the integrity of log records in transit. Log records shall be transferred as soon as possible after creation.
Evidence	Design documentation identifying the mechanism implemented for export of logs, including details of protection of logs against modification and details of format of log entries to facilitate integration into centralised logging and analysis.

DEV.604*: Record when device last seen

Criteria	This is applicable to all devices.
Constraints	A device (such as a controller) that has contact with other devices must be able to identify when it last had contact with another device. Where a device has not been seen for a period above a preset (possibly configurable) limit, a log record must be generated identifying the device that has not been seen. The trigger limit is likely to vary depending on the type of device and appropriate periods of inactivity.
Evidence	Design documentation identifying the mechanism implemented to detect and record contact with other connected devices, and the behaviour when contact is lost.

3.1.7 Development >> Cloud Services (External)

DEV.700*: Suitable cloud services

Criteria	This is applicable to all devices that use cloud services.
Constraints	If the product uses external cloud services, the developer must state how they meet the NCSC Cloud Security Principles as defined in the NCSC Cloud security guidance [Cloud]. The cloud service provider must have published their response to the NCSC Cloud Security Principles. *Note that in producing a Tailored Security Characteristic for a specific product evaluation, the evaluators shall include an identification of the services and assets that are to be deployed using external cloud services.*
Evidence	Design documentation stating how the product meets the NCSC Cloud Security Principles as defined in the NCSC Cloud security guidance [Cloud]. The cloud service provider's published response to the NCSC Cloud Security Principles.

3.2 Deployment mitigations

3.2.1 Deployment >> General

DEP.105*: Encrypt sensitive data

Criteria	See DEV.105.
Constraints	Sensitive data should not be stored on devices that are exposed outside of the secure enclave. If devices that contain sensitive data are removed from the secure enclave (e.g. for specialist analysis) then this must be done under procedural controls that minimise the specific risks to the deployment. Devices containing sensitive data must be configured to use the protection afforded by mechanisms such as BitLocker or equivalents. Refer to [EUD] for specific guidance for end-user devices.
Evidence	Installation, deployment or administration documentation identifying any configuration or administrative action required to ensure that functions of the host platform are used to protect sensitive data. Administration documentation identifying procedural controls to protect sensitive data that is removed from the secure enclave. (The list of sensitive data handled by the PSS is required as part of DEV.105 (Encrypt sensitive data).)

DEP.106*: Updateable product

Criteria	See DEV.106.
Constraints	For critical vulnerabilities the update must be applied within 14 days of the update becoming available. A software update mechanism must be implemented unless it is infeasible, in which case justification must have been provided for evaluation of DEV.106.
Evidence	The product's deployment guidance identifying where and how an administrator is to be made aware of update availability and how to obtain updates. Installation, deployment or administration documentation identifying any configuration or administrative action required to ensure that updates are regularly applied. Installation, deployment or administration documentation identifying any configuration or administrative action required to ensure that updates that are addressing critical vulnerabilities are applied within 14 days.

DEP.110*: Administrator authorised updates

Criteria	See DEV.107.
Constraints	A software update mechanism must be implemented unless it is infeasible, in which case justification must have been provided for evaluation of DEV.106 and DEV.107. The update process must provide a mechanism to ensure that updates are authenticated before they are applied. If this is not automatically checked by the update process, there must be confirmation by an administrator that the authenticity check has been successfully performed.
Evidence	Installation, deployment or administration documentation identifying any configuration or administrative action required to ensure that updates are authenticated by an authorised administrator before being applied.

3.2.2 Deployment >> Physical Security

DEP.200: Disable non-operational logical and physical interfaces

Criteria	This is applicable to all devices.
Constraints	See DEV.200.
Evidence	Installation, deployment or administration documentation identifying any configuration or administrative action required to manage non-operational interfaces.

DEP.201: Tamper response

Criteria	See DEV.201.
Constraints	If a device generates an alert it must be capable of being delivered and acted upon. See DEV.601.
Evidence	Installation, deployment or administration documentation identifying any configuration or administrative action required to ensure that tamper alerts are collected. Installation, deployment or administration documentation identifying any configuration or administrative action required to ensure that log entries are not lost by overwriting before they have been exported.

DEP.203: Protection of security-related physical structure

Criteria	See DEV.203.
Constraints	Use tamper evidence measures (as specified in [SC]) to make entry to system internals detectable by physical inspection. The deployment guidance must make it clear which devices need to be deployed in the secure area or secure enclave with appropriate physical protection.
Evidence	Installation, deployment or administration documentation identifying any installation or administrative action required to ensure that tamper evident measures are employed at access points on a device. Deployment or administration documentation providing guidance on examining tamper evident measures and identifying actions required to deal with detected tampering. Installation, deployment or administration documentation identifying which devices need to be deployed in the secure area or secure enclave with appropriate physical protection.

DEP.204: Physical security of management interfaces

Criteria	This is applicable to all devices.
Constraints	End-user devices that are employed to access management interfaces must not be accessible in a non-secure area. Admin access to subsystems that are deployed within the secure enclave, must also be within the secure enclave. Admin access to subsystems that are deployed outside the secure enclave but within a secure area, may be within the same secure area.
Evidence	Installation, deployment or administration documentation identifying which devices need to be deployed in the secure area or secure enclave, with end-user devices for accessing management interfaces on those devices deployed within the same area.

3.2.3 Deployment >> Secure Configuration

DEP.300: Provide a configuration tool to enforce required settings

Criteria	This is applicable to all devices.
Constraints	See DEV.300.
Evidence	Installation, deployment or administration documentation advising the administrator to perform the initial configuration using a supplied tool, policy template, or specific configuration guide to achieve this in as few steps as possible. The documentation must also include an effective method for the administrator to check that the deployment is in the evaluated configuration. The deployment guidance needs to cover all relevant aspects for the particular PSS, such as use (or disabling) of wireless communications where applicable, and how to recognise breach of the tamper boundary and visible changes related to DEV.200 and DEV.201.

DEP.302: Ensure product security configuration can be backed up

Criteria	This is applicable to all devices.
Constraints	See DEV.302.
Evidence	Installation, deployment or administration documentation advising the administrator to use the product's features to securely backup their configuration. Installation, deployment or administration documentation providing guidance to the administrator on the process of restoring the security configuration in a timely fashion in the event of a failure.

DEP.303*: Deploy onto suitably protected endpoint

Criteria	(Expected at least for end user devices)
Constraints	If the endpoint device is not provided with the product, the relevant security guidance for end-user devices provided at [EUD] must be followed where possible.
Evidence	Installation, deployment or administration documentation identifying any configuration or administrative action required to ensure that the endpoint is configured in line with good IT practice, equivalent to [EUD] guidance.

3.2.4 Deployment >> Network Security

DEP.401: Wireless network must be secured

Criteria	This is applicable to all devices.
Constraints	See DEV.401. Wireless technologies must not be used on any site requiring more than a basic level of protection.
Evidence	Installation, deployment or administration documentation identifying any configuration or administrative action required to ensure that suitable security mechanisms are employed to protect wireless communications channels.

DEP.402: Use whitelist to limit communications

Criteria	This is applicable to all devices.
Constraints	See DEV.402.
Evidence	Installation, deployment or administration documentation identifying any configuration or administrative action required to configure and use the whitelisting feature, including procedures for authorising devices to communicate.

DEP.403*: Use time synchronisation

Criteria	See DEV.403.
Constraints	See DEV.403.
Evidence	Installation, deployment or administration documentation identifying any configuration or administrative action required to configure time synchronisation. If this is not part of the product the deployment guidance must provide advice on how this can be implemented and configured. Installation, deployment or administration documentation identifying any configuration or administrative action required to ensure that the time protocol in use is still supported, and that all up to date security patches have been applied.

DEP.404*: Use segregated networks

Criteria	See DEV.404.
Constraints	See DEV.404.
Evidence	Installation, deployment or administration documentation identifying any configuration or administrative action required to configure segregated networks. If the device is not supplied with network setup, guidance must be provided to enable the administrator to ensure that network segregation is employed.

DEP.408: Do not deploy wireless technology at sites requiring more than basic protection

Criteria	This is applicable to all devices.
Constraints	Wireless networks must not be used on any site requiring more than a basic level of protection.
Evidence	Installation, deployment or administration documentation that advises the administrator not to employ wireless networks on a site requiring more than a basic level of protection.

3.2.5 Deployment >> Authentication Management (Privileges)

DEP.500*: Role based access control

Criteria	(Expected at least for end user devices)
Constraints	See DEV.500.
Evidence	Installation, deployment or administration documentation that identifies each role available for an administrator to assign to users, making clear what each role allows to be performed. This would be expected to include advice that users are assigned a specific role appropriate to the functions they need to perform.

DEP.501*: User least privilege

Criteria	(Expected at least for end user devices)
Constraints	See DEV.501.
Evidence	Installation, deployment or administration documentation that identifies the (OS and/or product-defined) privileges required for each user role. Deployment guidance to the system administrator to ensure that unnecessary privileges are not assigned to users.

DEP.502*: User authentication

Criteria	(Expected at least for end user devices)
Constraints	See DEV.502.
Evidence	Installation, deployment or administration documentation that identifies the user authentication options and how they are configured. Deployment guidance to the system administrator to ensure that a password policy is defined to be at least as robust as that defined in Appendix C of [SC]. Deployment guidance to the system administrator to ensure that password change is enforced upon suspicion that a password has been compromised. Deployment guidance to the system administrator to ensure that all default passwords are changed at installation to passwords that comply with the password policy. The guidance should include the warning that the deployment will not be CAPSS compliant if any default passwords have not been changed.

DEP.503*: One administrator per account

Criteria	(Expected at least for end user devices)
Constraints	The deployment must use one admin account per administrator.
Evidence	Deployment guidance to the system administrator to ensure that each administrator is provided with a separate user account, and no administrative user account is shared by multiple users.

DEP.504*: Local management authentication

Criteria	(Expected at least for end user devices)
Constraints	See DEV.504.
Evidence	Installation, deployment or administration documentation that identifies the user authentication options and how they are configured. Deployment guidance to the system administrator to ensure that administrative users are required to use MFA authentication.

DEP.505*: Remote management authentication

Criteria	(Expected at least for end user devices)
Constraints	See DEV.505.
Evidence	Installation, deployment or administration documentation that identifies the remote management access options and how they are configured. This should include reference to relevant expert guidance such as [RM_NCSC], [IPsec_NCSC] and [TLS_NCSC]. Deployment guidance to the system administrator to ensure that remote management access is protected by the choice of a secure protocol such as IPsec, SNMPv3, TLS or SSH with MFA authentication.

3.2.6 Deployment >> Monitoring

DEP.600*: Log all relevant events

Criteria	This is applicable to all devices.
Constraints	See DEV.600.
Evidence	Installation, deployment or administration documentation that identifies the logging mechanism and how it is configured. Where the events to be logged are configured by an admin user, the deployment guidance must include information on how to configure the product to ensure that the events logged include as a minimum those identified for DEV.600. Deployment guidance to the system administrator to ensure that, where available, the logs should be automatically exported to a management device in a secure area. Deployment guidance to the system administrator to advise that logs are analysed in a timely fashion and impact of unexpected entries assessed, following established organisation procedures for incident resolution.

DEP.602*: Export logs

Criteria	This is applicable to all devices.
Constraints	See DEV.602.
Evidence	Installation, deployment or administration documentation that identifies the logging mechanism and provides guidance on how to configure it for automatic export and to ensure integrity of logs in transit. Deployment guidance to the system administrator to advise that the logs should be automatically exported to a management device in a secure area. Deployment guidance to the system administrator to advise that integrity of logs is protected in transit.

DEP.603*: Audit log review

Criteria	This is applicable to all devices.
Constraints	See DEV.600.
Evidence	Deployment guidance to the system administrator to advise that logs are analysed in a timely fashion and impact of unexpected entries assessed, following established organisation procedures for incident resolution.

DEP.605*: Synchronised event time-stamps

Criteria	This is applicable to all devices.
Constraints	All event time-stamps must be synchronised with a reliable time source.
Evidence	Installation, deployment or administration documentation that identifies how time sources are used to ensure that time stamps are accurate, and provides information on configuration options for setting up or connecting to reliable time sources. Deployment guidance to the system administrator to advise that a reliable time source must be established for event time stamps, including guidance on how to assure reliability of a time source.

3.2.7 Deployment >> Cloud Services (External)

DEP.700*: Suitable cloud services

Criteria	This is applicable to all devices that use cloud services.
Constraints	See DEV.700.
Evidence	Installation, deployment or administration documentation that identifies how the product uses the cloud service provider's services and provides information and guidance on any action required to ensure that the configuration meets the NCSC Cloud Security guidance [Cloud].

Section 4 – Application Notes on Build Standard

In this section Application Notes are given for Build Standard requirements (from [BS]) that may require adaptation and interpretation to cover PSS devices. A general discussion is given in section 4.1 about the need to apply the Build Standard requirements to hardware devices and their manufacturing processes. In later sections, the Application Note table format used for SC mitigations is applied to selected Build Standard requirements, to highlight adaptations and interpretations that should be used for PSS products.

In this section, the abbreviation 'Rn' (where 'n' is a number) is used to indicate 'Requirement n' in [BS]. Because each requirement in [BS] identifies a table with a number of sub-requirements, in this document elements of a larger Build Standard requirement are given a 'minor requirement number' that identifies a particular part of the requirement (or its Description or Assurance Activity text in [BS]), and a short title intended to act as a reminder of the general content of the minor requirement. For example, the Assurance Activity for R6 in [BS] requires that "The Evaluation Team must investigate the Developer's physical and logical protection of their configuration management system" and minor requirements 6.1 and 6.2 are therefore introduced in this document to separately identify "Physical protection of CM system" and "Logical protection of CM system" respectively.

4.1 Application of build standard to hardware and manufacturing processes

The Build Standard requirements are applicable to all parts of a product under evaluation. Although [BS] is defined in terms of software development requirements, PSS' will generally include physical components and a manufacturing process. The Build Standard requirements for control over development processes will therefore apply to at least some aspects of the manufacturing. For example:

- **Tamper protection mechanisms that depend on physical components (e.g. switches, sensors, tracks on specific PCB layers, potted components) will need to ensure that the correct components are used in the manufacturing process;**
- **Components may contain firmware that is relied on for some of the mitigations (e.g. encryption, memory management) and therefore specific versions of the components need to be used;**
- **Components may require injection of certain data, whether public data such as a unique identifier or authorised public key, or secret data such as a seed value or shared cryptographic key – this therefore needs reliable sourcing of the data, suitably controlled injection during the manufacturing process, and possibly secure storage of any associated data for use after manufacturing (e.g. storing generated identifiers or shared keys).**

Therefore all Build Standard requirements need to be extended to consider their relationship to the hardware elements and manufacturing process of the product. In particular this means:

- **Configuration Management (R1, R3, R4, R6, R8) needs to include identification of hardware components, including versioning of the hardware design and unique identification of the hardware components (e.g. through an automated Bill of Materials system).**
- **The Build process (R7) needs to include the manufacturing process, confirming that the relevant items (such as firmware images, configuration data, parts lists) are reliably transferred to the manufacturing environment and held securely there (to prevent potential unauthorised modification). Where sensitive data such as identifiers, public keys, secret/private keys or seeds are generated, injected and/or stored then the security of these aspects needs to be included in the scope of the analysis of procedures and the site audit. If devices are held in a sensitive state before delivery (e.g. if the device is in a special mode that may allow configuration functions that are not available in the final deployed state) then it may be necessary to examine the security processes to protect this stage of the manufacturing process.**
- **Vulnerability handling (R2, R5, R9, R12) needs to include vulnerabilities identified in hardware components and design.**
- **Testing (R10) needs to include testing of hardware mechanisms relied on for the mitigations. In addition, the manufacturing process will generally include a test stage before the component is accepted as ready for delivery. This test stage may need to check correct operation of security mechanisms such as tamper signals,**

or random number generation. The test stage may also need to confirm that features such as debug modes (cf. discussion of R11 below) have been successfully disabled.

- **Use of security features of the platform (R11) may include use of hardware features such as memory management units (to limit memory available to a component or process), watchdog timers, one-time-programmable memory, and memory write protection (i.e. to protect firmware in memory from unauthorised changes) that are present in components, in order to limit the potential impact of attacks. In addition there may be security-significant steps required in the manufacturing process to inject sensitive data, disable debug modes (possibly after they have been used for data or firmware injection), and write-protect memory.**

4.2 Requirement 6 – configuration management system protection

4.2.1 Requirement 6.1: Physical protection of CM system

Criteria	Software or firmware updates (DEV.106) and/or software or firmware for initial installation are issued with cryptographic signature to protect their integrity and authenticity.
Constraints	The keys used for protecting integrity and authenticity of updates must be protected against unauthorised physical access.
Evidence	Procedure documentation describing how the signing keys are held securely and made available to authorised staff only. Witnessing of the signing key location and its access controls.

4.2.2 Requirement 6.2: Logical protection of CM system

Criteria	Software or firmware updates (DEV.106) and/or software or firmware for initial installation are issued with cryptographic signature to protect their integrity and authenticity.
Constraints	The keys used for protecting integrity and authenticity of updates must be protected against unauthorised physical access.
Evidence	Procedure documentation describing how the signing keys are held securely and made available to authorised staff only. Witnessing of the update process (real, or simulated for evaluation purposes) and the correct application of access controls. Backup documentation and/or observation confirming that the signing key cannot be accessed from backups by unauthorised personnel.

4.2.3 Requirement 6.3: Availability protection of CM system

Criteria	Always applicable.
Constraints	The configuration management system must be protected against loss of availability that would prevent either reconstruction of earlier builds (cf. requirement 1) or building of new instances of the product, over an extended period.
Evidence	Evidence would typically be expected in the form of a business continuity plan and associated procedures at a level of detail that defines the measures applicable to the CM system (and any other critical parts of the build process). Normally this would include the definition of offsite backup procedures, and periodic confirmations that the offsite backups taken can be restored in the ways required to achieve business continuity. No specific maximum unavailability period is defined here, but the period expected would be of the order of days or weeks rather than months. (It is noted that different parts of the CM, build and manufacturing environments may have different periods applicable).

Appendix A – References

This document references the following resources.

Label	Title	Version	Date	Location	Reference
62443-4-1	IEC 62443-4-1:2018, Security for industrial automation and control systems – Part 4-1: Secure product development lifecycle requirements	1.0	January 2018	https://webstore.iec.ch/publication/33615	
62443-4-2	IEC 62443-4-2:2019, Security for industrial automation and control systems – Part 4-2: Technical security requirements for IACS components	1.0	February 2019	https://webstore.iec.ch/publication/34421	
BS	NCSC CPA Build Standard	1.4	Oct 2018	https://www.ncsc.gov.uk/information/commercial-product-assurance-cpa	NCSC-1844117881-312
CEPlus	NCSC Cyber Essentials Plus			https://www.cyberessentials.ncsc.gov.uk	
Cloud	NCSC Cloud security guidance		17 Nov 2018	https://www.ncsc.gov.uk/collection/cloud-security	
CMMC	Cybersecurity Maturity Model Certification (CMMC)	1.02	18 March 2020	https://www.acq.osd.mil/cmmc/docs/CMMC_ModelMain_V1.02_20200318.pdf	
Control_Room	CPNI Control Rooms Guidance		Dec 2016	https://www.cpni.gov.uk/system/files/documents/73/38/Control%20Rooms%20Guidance%20Dec%202016.pdf	
EUD	Mobile Device Guidance *(this was formerly the 'End User Device Security Collection')*	1.0	22 Jan 2020	https://www.ncsc.gov.uk/collection/mobile-device-guidance	
IEEE802.1X	IEEE Standard for Local and metropolitan area networks – Port-Based Network Access Control	2010	2010	https://standards.ieee.org/standard/802_1X-2010.html	
IPsec_NCSC	NCSC Guidance – using Ipsec to protect data		23 Sept 2016	https://www.ncsc.gov.uk/guidance/using-ipsec-protect-data	
ISO27001	Information Security Management Systems: Requirements	2013	2013	https://www.iso.org/isoiec-27001-information-security.html	

Label	Title	Version	Date	Location	Reference
ISO29147	*Information technology — Security techniques — Vulnerability disclosure*	2018	2018	https://www.iso.org/standard/72311.html	
ISO30111	*Information technology — Security techniques — Vulnerability handling processes*	2019	2019	https://www.iso.org/standard/69725.html	
ISO9001	Quality Management Systems: Requirements	2015	2015	https://www.iso.org/iso-9001-quality-management.html	
Map	Cyber Assurance of Physical Security Systems (CAPSS) – 2021 – Potential mappings from CAPSS requirements to other schemes	1.0	2021	Available from CPNI	
MISRA	MISRA C: 2012 – Guidelines for the use of C language in critical systems	2012	March 2013	https://misra.org.uk/	
MISRA Comp	MISRA Compliance:2020 – Achieving compliance with MISRA Coding Guidelines	2020	February 2020	https://misra.org.uk/	
Pwned_NCSC	Suitable list of compromised passwords			https://www.ncsc.gov.uk/static-assets/documents/PwnedPasswordTop100k.txt	
PPFGE	Process for Performing CPA Foundation Grade Evaluations	2.5	Oct 2018	https://www.ncsc.gov.uk/information/commercial-product-assurance-cpa	NCSC-1844117881-485
RM_NCSC	Protect your management interfaces		22 March 2017	https://www.ncsc.gov.uk/blog-post/protect-your-management-interfaces	
SC	Cyber Assurance of Physical Security Systems (CAPSS) – 2021 – Security Characteristic	1.1	18 June 2021	https://www.cpni.gov.uk/cyber-assurance-physical-security-systems-capss	
SP800-53	NIST Special Publication 800-53 – Security and Privacy Controls for Information Systems and Organizations	Revision 5	September 2020	https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r5.pdf	
SP800-63B	NIST Special Publication 800-63B – NIST Digital Identity Guidelines – *Authentication and Lifecycle Management*		June 2017 Including updates as of March 2020	https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-63b.pdf	

Label	Title	Version	Date	Location	Reference
SP800-171	NIST Special Publication 800-171 – Protecting Controlled Unclassified Information in Nonfederal Systems and Organizations	Revision 2	February 2020	https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-171r2.pdf	
TLS_NCSC	NCSC Guidance – using TLS to protect data		17 Dec 2017	https://www.ncsc.gov.uk/guidance/tls-external-facing-services	

Appendix B – Glossary

The following definitions are used in this document.

Term	Definition
AACS	Automated Access Control System
AACS Controller	Back office system which controls the AACS
Always applicable	See definition in section 2.3
Applicable if Present	See definition in section 2.3
APT	Advanced Persistent Threat – a targeted cyber attack where a sophisticated attacker with significant expertise and resources accesses a system and remains undetected for a long time.
CAPSS	Cyber Assurance of Physical Security Systems
CCTV	Closed Circuit Television
CNI	Critical National Infrastructure
Conditionally applicable	See definition in section 2.3
CPA	Commercial Product Assurance
CUI	Controlled Unclassified Information (US term used in [SP800-171] and [CMMC])
Device	A physically distinct part of a product. Some products may consist of only one device.
DoS	Denial of Service
Element	A physically or logically distinct part of a system. An element may consist of a device or software (or both).
IA	Information Assurance
IPsec	Internet Protocol Security
Low functioning device	A device such as an FPGA/ASIC device, a simple circuit, or a simple device with very minimal firmware.
MAB	MAC Authentication Bypass
MFA	Multi-Factor Authentication
NIST	The US National Institute of Standards and Technology
Non-secure area	An area that is not secured, such as public spaces and building exteriors.
NTP	Network Time Protocol
OS	Operating System
PIR	Passive Infrared
Product	The target of the evaluation. A product may consist of a single device, a subsystem or a system.
PTP	Precision Time Protocol, also known as IEEE 1588
SC	Security Characteristic
Secure area	A secured area with access limited to authorised personnel and escorted unauthorised personnel.

Term	Definition
Secure enclave	A secured area with access limited to individually authorised personnel, no unescorted access for unauthorised personnel, with records of access. Typically a secure server room or secure control room. See [Control Room] for guidance.
Security Characteristic	A standard which describes necessary mitigations which must be present in a completed product, its evaluation or usage, particular to a type of security product.
Sensitive data	Data which, if compromised, would undermine • the cyber security of the product, or • the physical security of the site (or of assets that the product is supposed to protect according to its own requirements or requirements in its intended deployment environment), or • a person's expectation of privacy. This includes personal data (related to the person and their expectation of privacy), configuration data and cryptographic material such as keys and passwords.
SNMP	Simple Network Management Protocol
SSH	Secure Shell
System	A group of related elements, especially when dedicated to a single application.
Subsystem	A self-contained system within a larger system.
TLS	Transport Layer Security
TSC	Tailored Security Characteristic
WPA2	Wi-Fi Protected Access II

Appendix C – Report Guidelines

This Appendix is not used in the current version of the document

Appendix D – Fuzzing Guidelines

(In the sections below the word 'shall' in a requirement means that that part of the requirement must be met in order for the relevant assurance to be gained and for the evaluators to conclude a 'Pass'. The word 'should' in a requirement means that if that part of the requirement is not met then a rationale shall be given for why this is not believed to affect the assurance in the product – it may be advisable to confirm such rationales with their Test Lab in advance of completing the testing. 'Should' requirements generally indicate areas in which it can be expected that requirements will be strengthened in the future.)

D.1 Approach to Fuzzing

Basic guidance on fuzzing is included in [PPFGE], and this appendix extends that guidance with more specific comments on the expectations of coverage and analysis of results when carrying out fuzzing in a CAPSS evaluation. As noted in [PPFGE], there is a subjective element in the choice of what fuzzed messages are generated, and in how results of fuzzing are analysed to identify behaviour that may be indicative of exploitable behaviour and vulnerabilities. This appendix therefore aims to introduce principles and guidance that will improve the consistency of fuzzing in terms of the scale of testing and the coverage of the interface.

Protocols supplied by the platform or another 3rd party system component may be excluded from the scope of fuzzing provided that the 3rd party component is sufficiently widely used[6] and that evidence is provided by the developer that the 3rd party platform/component has a demonstrable vulnerability handling process covering receipt of vulnerabilities (e.g. from security researchers and customers), correction of vulnerabilities, notification of affected users, and secure distribution of updates).

The fundamental intentions of fuzzing an interface are:

- To apply an automated approach to generating test inputs that avoids using design analysis to create tests (and that therefore complements other testing that is specifically intended to be based on design)
- To achieve comprehensive coverage of protocol aspects that affect security
- To maximise expected coverage of security-relevant code paths
- To maximise coverage of attack surfaces available to attackers in the deployment environment for the target PSS
- To thereby identify behaviour of the target device that might indicate weaknesses that could be found in future to lead to exploitable vulnerabilities.

[6] Judging that a component is sufficiently widely used may be based on general recognition of its widespread deployment in contexts where bugs and vulnerabilities are likely to be noticed and reported (e.g. for operating systems such as Windows, or for components that are known to be widely deployed and analysed such as Mifare cards from a recognised manufacturer). In other cases the developer of the CAPSS target may be able to give evidence of the widespread deployment based on its use in their own products and perhaps their own acceptance testing or internal testing. In other cases the CAPSS judgement may be based on evidence from the manufacturer of the component.

The remainder of the guidance in this appendix relates primarily to structured fuzzing. Any unstructured fuzzing that is carried out (see [PPFGE, Annex A]) should answer the same analysis questions as in D.2 below, but does not need to meet the requirements in this subsection for coverage and content of the fuzzed messages.

For CAPSS purposes structured fuzzing is generally required at least for all unauthenticated aspects of the available interfaces. This shall therefore include the following:

- Handshake and key establishment messages, and messages used to establish authentication. However, an interface may also include messages that can be sent at any time without authentication (e.g. to close a session, reopen a session, enquire on the state of a session, change keys, or to reauthenticate), and these must therefore also be included.
- If the protocol has an 'authenticated' state (e.g. reached after a successful cryptographic handshake) then fuzzing should include a demonstration that unauthenticated messages are not processed in that state. Since an interface may process at least some message fields in order to determine that the message should be rejected, fuzzing must include such fields.
- Coverage of all fields in a message that are processed before authentication (e.g. header fields, and MAC or signature fields).
- Malformations of individual fields based on the type of that field (e.g. binary fields interpreted as unconstrained binary sequences, byte fields interpreted as signed integers, string fields, bytes interpreted as bit masks, byte fields interpreted as enumerated data types).
- Malformations at the level of the message as a whole – e.g. malforming a valid message by progressively truncating it one byte at a time, or by random overwriting across field boundaries.
- Messages that break dependencies inherent in the protocol – this includes dependencies such as length values, sequence counters, nonces established earlier in a session, signatures, MACs, fixed values (e.g. padding).
- In order to ensure that fuzzing is applied 'deeper' into the processing of messages, fuzzing shall include messages that meet protocol dependencies, so that the messages are not rejected because of the broken dependency before the fuzzed field is processed[7].
- Messages that malform data that is protected by hash, MAC or signature, as well as making the protection data itself invalid[8]. This can be used as part of confirming that the protected fields are not processed until after the protection has been checked (i.e. if the error in the protected field is notified instead of, or in addition to, the error in the protection (or if some other evidence such as change of state shows that the field has been used) then this indicates undesirable processing of data that has failed an integrity/authenticity check).

[7] It might be argued that where a dependency relates to a property based on a secret value, such as a cryptographic signature, keyed hash or MAC, then creating malformed messages that meet the dependencies (i.e. have a valid signature, etc.) can only be achieved by an authorised attacker, and therefore can be excluded from the scope of CAPSS fuzzing. This may be acceptable depending on the PSS context (e.g. depending on its risk assessment for insider attacks), and should therefore be agreed with CPNI, ideally as part of the Assurance Plan, or else (if the issue is discovered at a later stage) by separate agreement. Note that in some cases it may be relevant to include some amount of structured fuzzing of authenticated parts of messages as part of demonstrating that the fuzzed fields are not processed before authentication is achieved (e.g. to contrast the error arising from a content malformation when the protection data is correct compared to when it is incorrect).

[8] Making the protection invalid might not use fuzzing techniques, in order to have an identifiable 'expected error' result to check. For example, the wrong key might be used for the MAC or signature.

- If the target device is intended to be constrained to implementing a subset of the potential protocol messages (e.g. if it implements only a particular profile) then the fuzzing shall include sending valid and malformed messages outside that subset.

The execution of fuzz testing shall include reception, processing and recording of the target device response and associated messages (e.g. asynchronous alert messages that might be sent as a result of device state and/or processing of fuzzed messages). Fuzz testing should also collect any available audit logs that record security events related to the SC requirements (cf. DEV.600). Where the target may send asynchronous alerts then fuzz testing should also collect such alerts, and should do this in a way that allows analysis to detect alerts that may be significantly delayed from the message that caused them[9].

Where separate application and transport protocols are used, then fuzzing the application protocol will often be of most interest because it tends to have the most direct security impact on the target application. However, the transport protocol also needs to be considered according to the PSS context. For example, the transport protocol may be significant in terms of potential denial of service attacks, and in some cases certain security features may be provided by the transport layer. In general, fuzzing of the application layer should be done independently of security measures in the transport layer[10] (e.g. if the transport protocol provides encryption services then fuzzing of the transport protocol should include coverage of unencrypted, invalid encrypted, and valid encrypted messages; the application protocol fuzzing should then assume correct encryption at the transport layer).

Where a target protocol includes other embedded protocols or notations/encodings (e.g. where certain message types include X.509 certificates or other ASN.1 objects) then those 'embedded protocols' should also be fuzzed, noting especially where they introduce potential for additional features that might not be visible at the level of the main protocol (e.g. nested payloads where a data structure in the payload may include instances of the same data structure, requiring recursive processing in the target thus introducing recursive test cases to be included in fuzzing). The fuzzing of the embedded protocol should consider the possibility that the embedded protocol might be implemented by a separate component (possibly from a third-party source) that is capable of processing more than the range of values intended to arise from the target protocol.

D.2 Analysing the Results of Fuzzing

The requirements in this subsection shall be met for both structured and unstructured fuzzing that is part of a CAPSS evaluation.

The evidence collected from the execution of fuzz tests shall be analysed and at least the following questions should be answered using the results:

- Were any undecodable responses to fuzzed messages received?
- Were any decodable but invalid responses to fuzzed messages received?

[9] For example, this might mean collecting alerts received throughout a test session, then subsequently identifying alert types that are of interest (e.g. perhaps certain error conditions, or conditions that should not have arisen during the test session) and looking for causes in preceding messages by manual or automated means.

[10] There might be exceptions to this approach if the transport layer can be demonstrated to provide all of the authentication required for the communication. This would generally not be a valid approach in an environment where, for example, any device can join the target network by obtaining a network key for the transport layer (perhaps even in unencrypted form) from a controller device.

- Were any valid/decodable responses to invalid fuzzed messages received?
- Were any overlength fields returned (this is generally a subset of 'invalid responses received')
- Were any positive indications of device crash, hang or reset observed? (Examples of such indications could be sequences of messages ignored by the device, indicative messages in the application or transport protocols (e.g. re-establishing identifiers/addresses/links), or timeout waiting for an expected response)
- Were any fuzzed messages with malformed message protection (e.g. signatures, MACs) accepted by the device?
- Were any fuzzed messages with other malformed content accepted and actioned by the device?
- Were any messages outside the claimed subset accepted and actioned by the target?
- Did decryption of any encrypted response (field) fail?
- Were any security-related log events observed without a corresponding cause in the fuzzing (or in the environment)?
- Were there any alerts or responses missing a corresponding log event? (E.g. where an authentication failure arose, or a fuzzed message caused another event covered by DEV.600)

The evaluators should also create a 'watchlist' of events that should never (or rarely) arise during the testing, and scan the collected evidence (e.g. logs and asynchronous alerts) for these. Any watchlisted events found should then be investigated and reported.

Some results found during fuzzing (and from answering the analysis questions above) may not directly indicate a security vulnerability, but in general any unexpected behaviour (or behaviour leading to non-conformance with the interface specification) should be considered using the question "Why should this not be corrected before certification of the product?"

The fuzzing shall be deemed to have failed if any of the following results are observed on the target product in its evaluated configuration:

- The targeted interface stops working and requires human intervention to restart it (this includes cases where the target product hangs (still executing but not responding) or crashes)
- The targeted interface exhibits uncontrollable and unexpected dropping of messages
- The behaviour of the targeted interface prevents necessary emergency actions (e.g. where an unacceptable, out-of-specification delay in responses for critical messages is observed)
- The target product returns additional (unexpected) data
- The target product ignores an authentication or integrity failure (e.g. accepts a message with invalid MAC or signature)
- The target product suffers demonstrable corruption of state or logical integrity
- This list is not exhaustive, and represents a minimum list of generic failure indicators: other cases where security requirements of the target are breached (possibly indicated by other analysis questions above) would also be cause for failure.

Sometimes the results of fuzzing may demonstrate failure of conformance by the target to a claimed interface specification, although without an obvious security impact. Any such instances shall be reported by the evaluator. These non-conformances may be sufficient to lead to failure of certification (because they indicate a failure of the device to conform to specification, which in turn

casts doubt on the effectiveness of validation measures in the development environment[11]). Exceptions may be made by agreement with CPNI where the non-conformance can be shown not to adversely impact the security requirements.

D.3 Evidence to be Reported

The evaluation report shall identify any fuzzing evidence that was supplied by the developer, and how this was used by the evaluators (see D.4 below).

The approach used to generate fuzzed messages shall be described and a short rationale given as to how the requirements in D.1 have been met[12].

In order to demonstrate sufficient coverage of the protocol, the evaluators' reported results of structured fuzzing shall include description of the coverage of the protocol that has been achieved by the fuzzed messages in terms of:

- Total number of fuzzed messages sent
- Distribution of fuzzed messages across each included message type (message types may vary and may include, for example, syntactic message types defined by content, protection message types defined by different protective measures applied, and state-base message types defined by the messages accepted in each connection state)
- Distribution of fuzzed messages across each pre-authentication message type
- Distribution of fuzzed messages across each pre-authentication field (e.g. header plus MAC or signature fields, according to which fields are processed before authentication for the particular protocol)
- Malformation types applied.

The reporting of coverage shall also include a rationale that the fuzzed messages used provide sufficient coverage of the interface to make it unlikely (though not impossible) that fields available to a remote unauthenticated attacker have not been covered. For example, the rationale might be based on:

- Description of the generation algorithms used to create the test messages (explaining how this ensures that all relevant messages and fields are covered, and how it prioritises high risk fields and values)
- The numbers of fuzzed messages of each message type; demonstrating that all message types have been covered
- Description of the approach to any nested/recursive content that exists in the target protocol.

Where unstructured fuzzing is used (or where a choice is made that parts of an interface that do not require structured fuzzing are in fact subject to structured fuzzing with coverage that is more superficial than that required above for pre-authentication messages/fields), then a rationale shall

[11] Depending on the nature of the non-conformance, this may in turn lead to (re-)examination of relevant Build Standard requirements such as the pre-requisite in [BS, para 10] that "The Product Developer must perform extensive testing of their products" and/or Requirement 10.

[12] This rationale may be combined with the reporting of coverage of the protocol, but is intended to ensure that the report as a whole describes not only *what* coverage was achieved but *how* it was achieved, in a way that enables a reader to understand that the claimed coverage would indeed result from the generation approach.

be provided to justify that the authentication check will cause failure (for messages without valid authentication) before the other parts of the message are processed. This rationale may be based on references to source code subsets that are provided to the evaluators.

Evidence should include summary tables for each of the answers to analysis questions (see D.2).

Developers should provide design evidence related to the processing of messages to identify all content that is processed before authentication (for any and all commands). Particular attention is drawn to providing evidence to support a rationale that fuzzing has covered all relevant fields processed before authentication (especially where the authentication is performed in an application protocol).

D.4 Using Fuzzing Evidence from the Developer

Fuzzing evidence can be supplied by the developer, and may be examined by the evaluators as a replacement for, or in addition to, fuzzing carried out by the evaluators. The requirements set out for fuzzing in the other sections of this Appendix (and the referenced parts of [PPFGE]) shall in that case be met by the combination of the fuzzing carried out by developer and evaluators.

When the developer supplies fuzzing results then this must include the raw data containing the fuzzed messages generated, and contemporaneous records of the execution of the fuzzing detailing the messages sent, responses collected, and any relevant alerts, audit logs, etc. (cf. D.1 above). The developer should also supply a short rationale as to how the requirements in D.1 (or a relevant subset of them) have been met. The evaluators shall examine any such rationale provided to in order to identify any potential inconsistencies with the observed results, and to identify any potential gaps or weaknesses that need to be addressed by additional evaluator fuzzing – the evaluators shall report the results of this examination.

If the developer does not supply analysis of the fuzzing results then the analysis required for CAPSS shall be done by the evaluators. If the developer supplies analysis, then the evaluators may use this analysis (supplemented by their own analysis in order to fill any gaps in the requirements for CAPSS fuzzing). The evaluators shall perform an independent examination of any analysis supplied by the developer, in order to generate confidence that the method of analysis is understood and consistent with the requirements for CAPSS fuzzing. This independent examination would typically consist of:

a) checking that the different parts of the developer's analysis are consistent (e.g. checking that numbers of messages sent, and coverage of messages/fields claimed is consistent between different parts of the analysis and different questions); and
b) checking a sample of the analysis questions to determine that consistent identifications and conclusions would be reached by the evaluators based on the raw results supplied.

Even if the developer supplies sufficient evidence to for all the CAPSS requirements, the evaluators may nonetheless decide to carry out some fuzzing themselves (either using developer-supplied tools or their own tools) as part of their activity to check the developer's evidence. Where such evidence has not been accepted as part of a CAPSS evaluation before, then the evaluators *should* carry out at least some independent fuzzing.

The evaluation report shall identify what evidence was supplied by the developer, and how this was used by the evaluators.

Appendix E – Changes from CAPSS 2019

This appendix describes the changes made in the Application Notes starting from CAPSS 2019 as the baseline.

E.1 CAPSS 2019 to CAPSS 2021 v1.0

(Note that no separate CAPSS 2020 version was published.)

The approach to determining relevant mitigations for a component has been revised in CAPSS 2021. The intent has generally been to maintain the same requirements but to make it clearer how to determine relevant mitigations for each target part of the PSS. This high-level change has the following elements:

- The 'Applicability' row has been deleted from all tables in the Application Notes – the approach to determining applicability of mitigations is now based on using the list in CAPSS 2019 SC Appendix D and some guidance on determining the applicability of a mitigation that has been added in section 2.2.
- The Criteria row for some requirements now uses the phrase "Expected at least for end user devices" to clarify that expectation (as described in section 2.2).
- References to variants have been removed, since the new approach does not use these.

The term 'highly constrained device' has been replaced with 'low functioning device' (previously both were used with effectively the same meaning).

Additional guidance on alternative standards that can be used to satisfy the pre-requisites relating to ISO 27001 and CyberEssentials Plus has been added in section 1.7.

Clarification of the authorship of CPA Security Procedures for CAPSS evaluations has been added in section 1.8.

Future changes that are anticipated to CAPSS requirements have been identified in section 1.10, in order to give advance warning to manufacturers and test labs.

Updates to requirements in the CAPSS SC have been reflected in changes to the relevant application notes in this document. Detailed changes are not described here, except sometimes to identify a high level theme, or where the change to application notes seems worth noting in its own right.

The definition of sensitive data, and its use in DEV.105 has been updated.

Additional guidance on the use and evaluation of MISRA has been added in DEV.108.

The application notes for DEV.200 and VER.200 have been updated to reflect the modification in the SC to identify alternatives to MFA.

Updates have been made (e.g. in DEV.201) to clarify that other high functioning devices (such as servers) generally need to implement equivalent measures to those in [EUD].

Additional guidance on evaluation activities has been added, especially for cases where wireless must be disabled in the CAPSS evaluated configuration (e.g. VER.401, DEP.401).

Additional guidance on MFA for local and remote management has been added in DEV.504 and DEV.505.

Additional guidance on fuzzing (VER.407) has been added in Appendix D.

An additional interpreted requirement on availability of the developer's CM system has been added as Requirement 6.3 in section 4.2.3.

References have been updated, including replacing the previous NCSC guidance on 'End User Devices' with guidance for 'Mobile Devices'. However, the term 'End User Device' has been retained for CAPSS 2021.

E.2 CAPSS 2021 v1.0 to CAPSS 2021 v1.1

Additional interpretation has been added to describe the potential use of CMMC as a pre-requisite in section 1.7.3, and updates have been made to clarify interpretations for DEV.300, DEV.504, DEV.505 and DEP.300 in CAPSS 2021 v1.1. No changes were made to the SC mitigations.