TECHNICAL SPECIFICATION

ISO/IEC TS 9569

First edition
2023-11

Information security, cybersecurity and privacy protection — Evaluation criteria for IT security — Patch Management Extension for the ISO/IEC 15408 series and ISO/IEC 18045

Sécurité de l'information, cybersécurité et protection de la vie privée — Critères d'évaluation pour la sécurité des TI — Extension pour la gestion des correctifs concernant la série ISO/IEC 15408 et l'ISO/IEC 18045

Reference number
ISO/IEC TS 9569:2023(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives or www.iec.ch/members_experts/refdocs).

ISO and IEC draw attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO and IEC take no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO and IEC had not received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at www.iso.org/patents and https://patents.iec.ch. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see www.iso.org/iso/foreword.html. In the IEC, see www.iec.ch/understanding-standards.

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 27, *Information security, cybersecurity and privacy protection.*

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html and www.iec.ch/national-committees.

Introduction

The ISO/IEC 15408 series is intended to be used to evaluate the assurance of IT products. While the ISO/IEC 15408 series can be used to perform an initial evaluation of an IT product, it does not support a differential security evaluation of that product, subsequent to one or several patches being applied to it. Neither the ISO/IEC 15408 series nor ISO/IEC 18045 contain dedicated methods or evaluation activities which would support the evaluation of changes or updates.

Some of these aspects were addressed by users of the ISO/IEC 15408 series, in particular evaluation authorities, but also within the mutual recognition agreements (e.g. Common Criteria Recognition Arrangement). In many real-world use-cases, developers provide updated or patched target of evaluations (TOEs), but the effort to re-certify these versions has mostly been avoided.

This problem of patch management and its related components are missing from the current ISO/IEC 15408 series and ISO/IEC 18045. To address this problem, requirements and recommendations are needed on how to regain assurance of an updated target of evaluation in a standardized and widely accepted way e.g. in terms of effort and costs.

This document collects discussions and experience from the experts involved in the ISO/IEC 15408 series and ISO/IEC 18045, to address the evaluation of the patch management during the evaluation of the initial TOE in a standardized way. This document also discusses alternatives for the evaluation of patched TOEs, although it does not provide a standardized approach.

This document is intended to be used as an extension to the ISO/IEC 15408 series and ISO/IEC 18045.

Clause 5 includes the definition of the new patch management assurance family following the structure defined in the ISO/IEC 15408 series and ISO/IEC 18045. Clause 6 includes additional guidance for the evaluators of the initial target of evaluation (TOE). Annex A summarizes experiences in evaluation schemes as options for adoption.

NOTE This document uses bold and italic type in some cases to distinguish terms from the rest of the text. The relationship between components within a family is highlighted using a bolding convention. This convention calls for the use of bold type for all new requirements. For hierarchical components, requirements are presented in bold type when they are enhanced or modified beyond the requirements of the previous component. In addition, any new or enhanced permitted operations beyond the previous component are also highlighted using bold type. The use of italics indicates text that has a precise meaning. For security assurance requirements, the convention is for special verbs relating to evaluation.

This document follows the conventions introduced in the ISO/IEC 15408 series and ISO/IEC 18045.

Information security, cybersecurity and privacy protection — Evaluation criteria for IT security — Patch Management Extension for the ISO/IEC 15408 series and ISO/IEC 18045

1 Scope

This document specifies patch management (PAM) security assurance requirements and is intended to be used as an extension of the ISO/IEC 15408 series and ISO/IEC 18045.

The security assurance requirements specified in this document do not include evaluation or test activities on the final target of evaluation (TOE), but focus on the initial TOE and on the life cycle processes used by manufacturers. Additionally, this document gives guidance to facilitate the evaluation of the TOE, including the patch and development processes which support the patch management.

This document lists options for evaluation authorities (or mutual recognition agreements) on how to utilize the additional assurance and additional evidence in their processes to enable the developer to consistently re-certify their updated or patched TOEs to the benefit of the users. The implementation of these options using an evaluation scheme is out of the scope of this document.

2 Normative references

There are no normative references in this document.

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at https://www.electropedia.org/

3.1
activation
operation performed on a patch to transform the *initial target of evaluation (TOE)* (3.8) into the *final TOE* (3.4)

Note 1 to entry: Activation is an atomic operation which can only be done in one step (partial activation is not allowed).

Note 2 to entry: In addition to installing the modified functionality, this operation shall encompass a change in TOE identification.

Note 3 to entry: The TOE shall remain in a secure state even if interruption or incident occurs during such operation, which prevents the forming of the final TOE.

3.2
end-of-support
date until when the user can expect to receive new patches

Note 1 to entry: The end-of-support should be greater than the period of validity of the certificate.

Note 2 to entry: The period of validity of the certificate can be extended through the standard assurance continuity.

3.3
evaluation authority
body operating an evaluation scheme

[SOURCE: ISO/IEC 15408-1:2022, 3.40]

3.4
final target of evaluation
final TOE
initial TOE (3.8) with the *patches* (3.11) applied

Note 1 to entry: The final TOE is obtained by combining the initial TOE and patch(es) to be loaded and activated on the initial TOE.

Note 2 to entry: The final TOE is not necessarily evaluated but assurance is gained through ALC_PAM on the initial TOE.

3.5
flaw remediation
assurance family ALC_FLR which provides requirements for the handling of security flaws

Note 1 to entry: This definition of flaw remediation is based on ISO/IEC 15408-3:2022, 12.1.

3.6
identification data
data that identifies the *initial target of evaluation* (3.8), the applied *patch(es)* (3.11) or the *final target of evaluation* (3.4)

3.7
initial evaluation
complete evaluation of the *initial target of evaluation* (3.8)

3.8
initial TOE
initial target of evaluation
target of evaluation (TOE) (3.18) that supports evaluated features allowing at least to securely load, activate and execute patch(es), without any applied patches

Note 1 to entry: The *final TOE* (3.4) is obtained by loading and activating the patches for the initial TOE.

Note 2 to entry: The final TOE may not be evaluated but assurance is gained through the evaluation of ALC_PAM on the initial TOE.

3.9
loader
piece of the *target of evaluation security functionality* (3.19) of the *initial target of evaluation* (3.8) that implements the *activation* (3.1) of a *patch* (3.11)

3.10
maintenance
process provided by an evaluation authority that recognises that a set of one or more applied *patches* (3.11) made to an *initial target of evaluation (TOE)* (3.8) has not adversely affected the assurance

Note 1 to entry: Changes in the development environment can be considered as maintenance if they relate to the TOE.

Note 2 to entry: Maintenance is typically applied in the context of certification.

3.11
patch
type of source code or binary code to be added to an *initial target of evaluation (TOE)* (3.8) in order to introduce additions or modifications of a functional or security feature

Note 1 to entry: A patch is loaded on the initial TOE and activated to obtain the final TOE.

Note 2 to entry: Full replacement of a TOE is a possible implementation of "patchability" and a current practice for software TOEs.

3.12
patch management
PAM
processes applied during *patch* (3.11) development and patch release

3.13
patch management documentation
PMD
documentation describing the policies, processes, procedures related to the patching of the *target of evaluation* (3.18)

3.14
patch verification mechanism
technical mechanism to verify the integrity and/or authenticity of a *patch* (3.11)

3.15
re-evaluation
process of recognising that changes made to an *initial target of evaluation* (3.8) require independent evaluator activities to be performed in order to establish a new assurance baseline

Note 1 to entry: Re-evaluation seeks to reuse results from a previous evaluation.

3.16
security assurance requirement
SAR
security requirement that refers to the conditions and processes for the development and delivery of the *target of evaluation* (3.18), and the actions required of evaluators with respect to evidence produced from these conditions and processes

[SOURCE: ISO/IEC 15408-1:2022, 3.76]

3.17
security relevance report
SRR
document containing the assessment of security relevance of a *patch* (3.11)

3.18
target of evaluation
TOE
set of software, firmware and/or hardware possibly accompanied by guidance, which is the subject of an evaluation

[SOURCE: ISO/IEC 15408-1:2022, 3.90]

3.19
target of evaluation security functionality
TOE security functionality
TSF
combined functionality of all hardware, software, and firmware of a *target of evaluation (TOE)* (3.18) that is relied upon for the correct enforcement of the security functional requirements

[SOURCE: ISO/IEC 15408-1:2022, 3.92]

3.20
transport
process of transferring patches from the developer to the user who applies the *patch* (3.11)

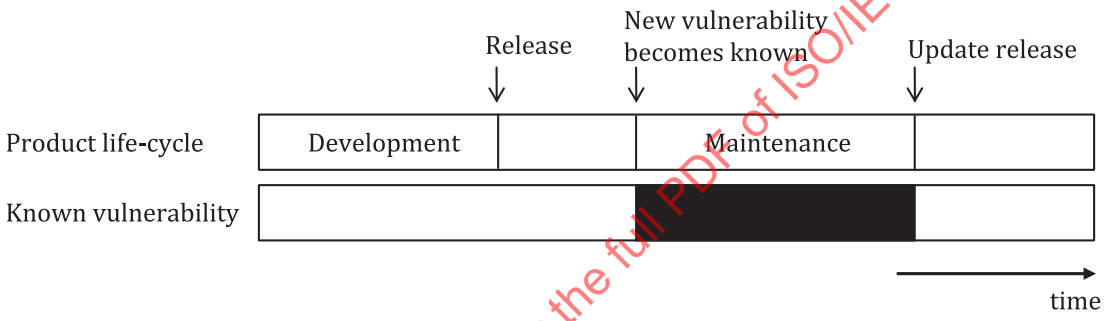
3.21
vulnerability
weakness in the *target of evaluation* (3.18) that can be used to violate the security functional requirements in a specified environment

Note 1 to entry: In the definition of ALC_PAM.1 in 5.2.4, the term flaw is used to ensure consistency with ALC_FLR components.

4 Overview

4.1 Background information

Figure 1 shows the product vulnerability timeline for the case after a new vulnerability is detected and becomes publicly known. Until the developer releases an update that removes the vulnerability, and that update is applied, the product will be insecure. This status is shown in black below.



Key

The product is vulnerable due to the lack of a patch.

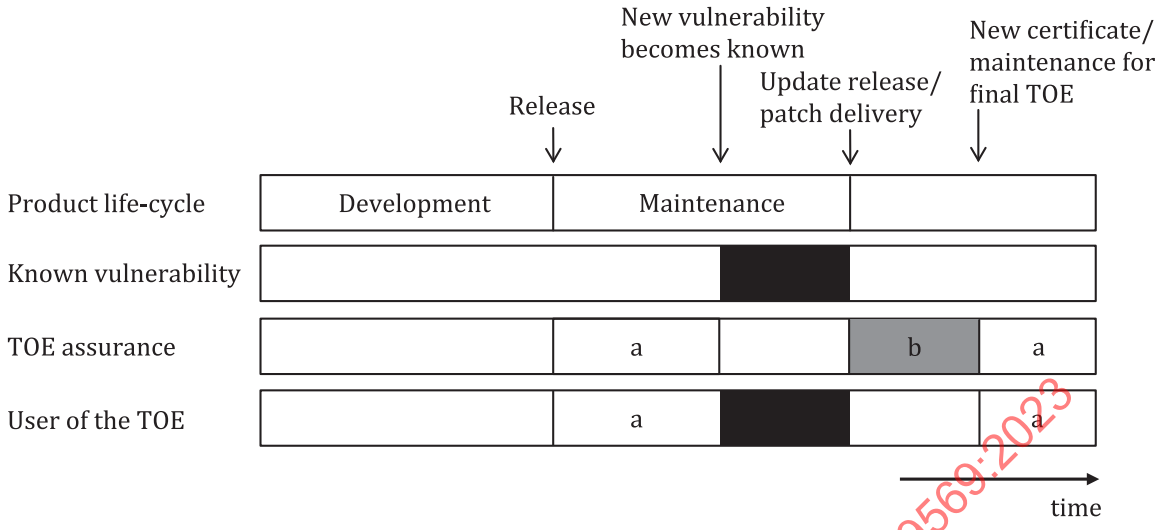
Figure 1 — Product vulnerability timeline

Consequently, developers have a responsibility to build and release those updates in a short period of time after the vulnerability becomes known. Developers who obtained a certificate previously may request a re-evaluation of the TOE (for example for issuing a new certificate, or because it is mandated by their clients). In many real-world cases, re-evaluation does not happen for every patch of the product, mostly due to cost and delay.

Since the patched TOE has not been re-evaluated, the developer can introduce a regression defect while deploying the vulnerability fix or in the fix itself. In the absence of evaluation by a skilled third party, there is a general lack of assurance on the patched TOE. This transfers the decision to use either a previously certified or a recently patched version to the user of the TOE.

Therefore, the user of the TOE should run their own risk assessment to determine which version of the TOE to use. If users of the TOE limit themselves to evaluated versions, they therefore accept known vulnerabilities in the TOE. Further risk mitigation should also be done, i.e. additional compensating countermeasures against the new vulnerabilities should be implemented. Conversely, using patched TOEs can also include flaws introduced by the developer during the patch development or deployment.

Figure 2 illustrates the timeline and relationship of a TOE when a new vulnerability occurs, a patch becomes available and the status of the certification is not in sync.



Key

■	The TOE is vulnerable due to the lack of a patch.
▭	The user is unable to decide whether it is better to use the evaluated TOE or the patched TOE.
a	The user can use the (re-)evaluated TOE.
b	Time for maintenance / re-certification.

Figure 2 — Timeline showing availability of patch and the corresponding new certificate

The focus is on the time for maintenance or re-certification (see Figure 2), in particular:

— how to ease re-evaluations, to optimally shorten the time for maintenance or re-certification;

— how to give some degree of assurance to the user so that, during this maintenance or re-certification period, they can choose to deploy the patched TOE.

This proposed patch management extension has the following advantages for the different stakeholders:

— Easing the re-evaluation process, therefore helping regulatory bodies in mandating re-evaluations when needed.

— Helping users to resolve the dilemma of whether to keep the evaluated version, or move to the patched version, by providing some degree of assurance on the patched TOE by assessing, during the initial evaluation that:

  — the patch deployment process provides procedural security measures against the introduction of regressions;

  — the TOE security functionality, including mechanisms allowing the TOE to be patched, are evaluated for conformity and robustness to avoid introducing vulnerabilities on the TOE.

— Helping developers by providing a standard way to assess the security of their patch development and deployment processes, as well as standard requirements to define the patching capabilities of their products.

— Helping evaluation authorities with a set of options they can provide within their policies to the customers (i.e. developers) to offer flexible and modern evaluation approaches.

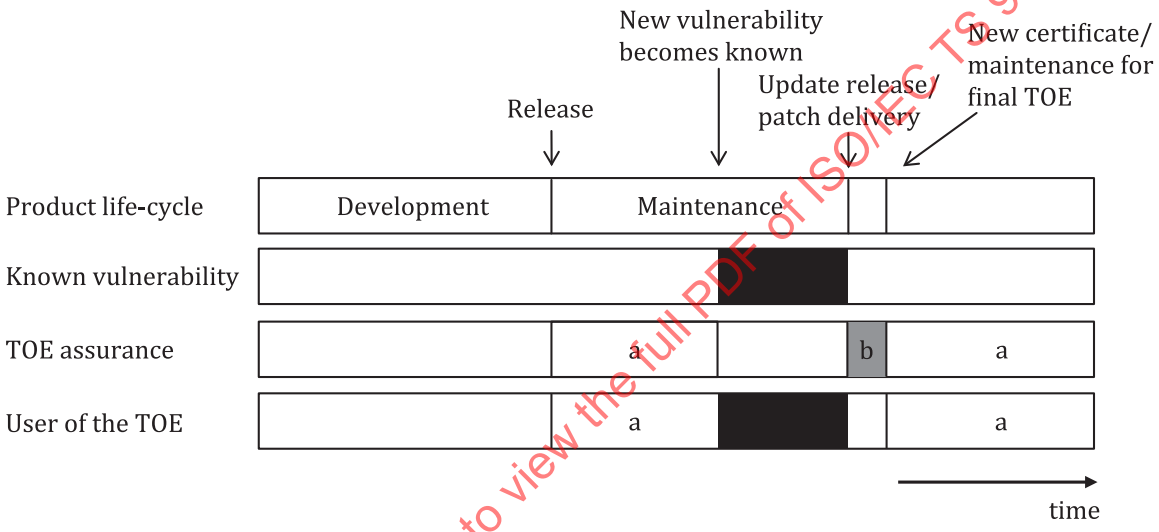
4.2 Proposed approach

The solution involves the following two aspects:

— Add additional functional requirements which address the patch or update functionality of the initial TOE. This document does not define mandatory content for the security problem definition or security functional requirements (SFRs). The security target or protection profile should contain TOE or TOE-type specific information. To facilitate the authoring of these documents, Annex C gives an example for a security problem definition and corresponding objectives. Additionally, Annex D includes guidance on how to write SFRs for the patch functionality.

— Add additional life cycle requirements (ALC_PAM) to get commitment from developers to consistently monitor for flaws or issues after release of the initial TOE, but also encourage developers to consistently generate evidence for future re-evaluations (see 5.2).

Figure 3 shows the application of ALC_PAM, which supports the timely delivery of the patch or update, but also the maintenance of the internal and external assurance activities.



Key

■	The TOE is vulnerable due to the lack of a patch.
■	The user is unable to decide whether it is better to use the evaluated TOE or the patched TOE.
a	The user can use the (re-)evaluated TOE.
b	Time for maintenance / re-certification.

Figure 3 — Timeline showing application of ALC_PAM

4.3 Non-public vulnerabilities

For many IT products, researchers discovering vulnerabilities are incentivised to not disclose the vulnerabilities until the developers have had an opportunity to patch them. In this case, it is plausible that the end user of the TOE is not aware of the vulnerability and the presence of the vulnerability can be considered a residual risk inherent to the use of any IT product. Consequently, many security patches are issued prior to end users and the public being made aware of the vulnerability.

The assurance family ALC_PAM introduced in this document provides a way to increase the assurance on developer patching procedures. When vulnerabilities are reliably fixed by patching procedures before the vulnerability is made public, there is less opportunity for successful attacks.

5 Patch management family

5.1 General

This clause defines the new assurance family ALC_PAM.

The security assurance requirements (SARs) introduced in 5.2 are related to different evaluation phases. During initial evaluation of the TOE, additional evaluation actions shall be introduced (compared to the standard SAR from ISO/IEC 15408-3) to establish assurance for the future patch generation process. The concept is to define ALC_PAM (patch management) and augment this family during initial evaluation in the security target.

As patch management is part of the life cycle assurance, it has been introduced under the ALC class. ALC_PAM describes how to handle patches life cycle, design, development, validation and release, but not the remediation flow. For this reason, ALC_PAM is not part of ALC_FLR (flaw remediation) even if a patch is a fix for a flaw managed in accordance with ALC_FLR. Both classes are closely related and therefore the dependency with ALC_FLR.2 was defined.

ALC_PAM, contrastingly, aims to support maintenance of the TOE assurance over the product life cycle. This family requires developers to provide a patch management policy and to follow this policy to develop patches for the TOE at the time of evaluation. This family also requires developers to define a procedure for the self-assessment to maintain the quality of the TOE after its evaluation. The developer can publish the result of the self-assessment to show the current status of the latest version of the TOE (e.g. re-evaluation is required or assurance is maintained) to the TOE users.

Annex B contains an example of a patch policy which fulfils the given requirements.

5.2 Patch management (ALC_PAM)

5.2.1 Objectives

The objective of this family is to identify the policies and procedures to be implemented in the development process, which will be applied after the initial release of a TOE by the developer.

The application of the patch management (PAM) process cannot be always determined at the time of the initial evaluation. Nevertheless, it is possible to evaluate the policies and procedures that a developer has in place to perform the PAM process for a future patch release. It is also possible to obtain some evidence of the correct application of the procedures during the patching of the problems which are found during the evaluation of other assurance classes like AVA (vulnerability assessment) and ATE (tests).

The written PAM policies, processes and procedures are internal documents for the developer. These shall include instructions, among others, on how developers securely provide guarantees of authenticity to distribute and apply patches and how the life cycle of the keys, used for providing authenticity of new patches, is handled.

These procedures shall guarantee the secure development, the secure deployment, installation and activation for patches. Moreover, the procedures and the set of commands supporting them shall be described in the AGD (guidance) family.

5.2.2 Component levelling

This family contains only one component.

5.2.3 Application notes

None.

5.2.4 ALC_PAM.1 Patch management

Dependencies: ALC_FLR.2 flaw reporting procedures.

Application note: The purpose of ALC_FLR is to build assurance of the flaw remediation procedures which are applied after security flaws were discovered. Separately, the purpose of ALC_PAM is to build assurance of the patch management processes which are applied when the behaviour of the initial TOE is changed independent of the type of change.

Therefore, the relationship of ALC_FLR to ALC_PAM is justified by the need to release patches to distribute flaw corrections.

Table 1 contains the developer action elements, Table 2 contains the content and presentation elements and Table 3 contains the evaluator action elements of ALC_PAM.1.

Table 1 — ALC_PAM.1 developer action elements

Element	Definition
ALC_PAM.1.1D	**The developer shall provide patch management documentation (PMD) for the TOE.**
ALC_PAM.1.2D	**The developer shall provide end-of-support information to the TOE users.**
ALC_PAM.1.3D	**The developer shall follow the PMD on a regular basis.**
ALC_PAM.1.4D	**The developer shall record evidence of the application of the PMD.**
ALC_PAM.1.5D	**The developer shall release patches as defined in the PMD until the end-of-support of the TOE.**
ALC_PAM.1.6D	**The developer shall follow the PMD to produce an updated set of evaluation evidence for each released patch at least until the stated end-of-support of the TOE.**
ALC_PAM.1.7D	**The developer shall provide a channel used to check for the availability and/or download of patches with means to protect the channel according to the specified security capabilities of the TOE.**
ALC_PAM.1.8D	**The developer shall create a security relevance report (SRR) for each patch release.**

Table 2 — ALC_PAM.1 content and presentation elements

Element	Definition
ALC_PAM.1.1C	**The PMD shall state the criteria used for the decision that a patch shall be released.**
ALC_PAM.1.2C	**The PMD shall require the generation of an SRR and shall identify any applicable procedure.**
ALC_PAM.1.3C	**The SRR shall describe the flaws, changes and impact that are related to the patch.**
ALC_PAM.1.4C	**The PMD shall describe how to update the initial TOE evidence for any applicable SAR.**
ALC_PAM.1.5C	**The PMD shall define how to record any PAM-related decision.**
ALC_PAM.1.6C	**The PMD shall describe the mandatory patch-specific content for the preparative procedures and the operational user guidance.**
ALC_PAM.1.7C	**The PMD shall describe the mandatory procedures during patch release.**
ALC_PAM.1.8C	**The PMD shall contain rules regarding testing (using internal resources or using external third party) before a patch is released.**
ALC_PAM.1.9C	**The PMD shall describe how end users are notified of a new patch and corresponding installation instructions.**
ALC_PAM.1.10C	**The PMD shall describe all necessary developer procedures to support the patch functionality of the TOE.**

Table 3 — ALC_PAM.1 Evaluator action elements

Element	Definition
ALC_PAM.1.1E	**The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.**

5.3 Evaluation work units for ALC_PAM

5.3.1 Action ALC_PAM.1.1E

5.3.1.1 General

ALC_PAM.1.1C *The PMD shall state the criteria used to decide that a patch shall be released.*

5.3.1.2 Work unit ALC_PAM.1-1

ALC_PAM.1-1 The evaluator ***shall check*** for the definition of the criteria which is used to decide that a patch shall be released, and check for the implementation as a policy. Example of a list of criteria:

— complexity of backports;

— operational stability, development teams are able to estimate effect for operational stability;

— security impact;

— customer impact (i.e. practical problems, theoretical problems);

— time impact, e.g. to address customer expectations;

— any other criteria dependent from developer business case.

5.3.1.3 Work unit ALC_PAM.1-2

ALC_PAM.1-2 The evaluator ***shall check*** the status of the implementation of the policies for patch releases and examine if such policies are detailed enough to enable a repeatable resolution of patch development, testing and release.

5.3.1.4 Work unit ALC_PAM.1-3

ALC_PAM.1-3 The evaluator ***shall examine*** if the following mandatory PMD content has been implemented:

— criteria used to decide that a patch shall be released;

— unique label for each patch to identify all release items.

ALC_PAM.1.2C *The PMD shall require the generation of an SRR and shall identify any applicable procedure.*

5.3.1.5 Work unit ALC_PAM.1-4

ALC_PAM.1-4 The evaluator ***shall check*** that the PMD mandates the generation of an SRR prior to patch release and that all the patching procedures are referenced unambiguously. If the policies distinguish between different categories of a patch, then the evaluator shall check that the SRR and the associated procedures cover each of the categories.

ALC_PAM.1.3C *The SRR shall describe the flaws, changes and the impact that are related to the patch.*

5.3.1.6 Work unit ALC_PAM.1-5

ALC_PAM.1-5 The evaluator ***shall check*** the format of the SRR used by the developer.

The SRR shall contain following mandatory elements:

— each flaw shall be listed and explained;

— the related changed shall be listed and explained;

— for each change, the security impact shall be given by means of security relevance criteria (e.g. remote execution, only product type specific) or a standardized category system [e.g. common weakness enumeration (CWE)].

Annex B includes a template for the SRR.

ALC_PAM.1.4C *The PMD shall describe how to update the evidence documentation used in the initial evaluation for any applicable SAR.*

5.3.1.7 Work unit ALC_PAM.1-6

ALC_PAM.1-6 The evaluator ***shall check*** if the PMD describes how to update the evidence documentation in a consistent way with the evaluation assurance level.

ALC_PAM.1.5C *The PMD shall define how to record any PAM-related decision.*

5.3.1.8 Work unit ALC_PAM.1-7

ALC_PAM.1-7 The evaluator ***shall check*** that the PMD describes how to record decisions related to the patch delivery.

ALC_PAM.1.6C *The PMD shall describe the mandatory patch-specific content for the preparative procedures and the operational user guidance.*

5.3.1.9 Work unit ALC_PAM.1-8

ALC_PAM.1-8 The evaluator ***shall check*** the PMD for instructions on how to update the initial TOE preparative procedures and operational user guidance anytime a patch is released. For example, by providing a checklist to cover all the steps of the patching process from loading to activation.

Application note: This work unit is different from ALC_FLR.2-5 because it requires developers to document how to update initial TOE documentation when a patch is released, and not how to notify users about how to fix a security flaw.

ALC_PAM.1.7C *The PMD shall describe the mandatory procedures during patch release.*

5.3.1.10 Work unit ALC_PAM.1-9

ALC_PAM.1-9 The evaluator ***shall check*** the PMD for mandatory patch release procedures.

ALC_PAM.1.8C *The PMD shall contain rules regarding testing (using internal resources or using external third party) before a patch is released.*

5.3.1.11 Work unit ALC_PAM.1-10

ALC_PAM.1-10 The evaluator ***shall check*** the PMD for rules that require different types of testing (e.g. by the evaluation facility, or by the developer) and what should be tested and how. For example, a rule set for the different roles in the (patch) release procedure such as development, quality assurance department, product owner, etc.

Evaluation authorities can define specific rules for the coverage and depth for re-testing until the TOE end-of-support.

ALC_PAM.1.9C *The PMD shall describe how end users are notified of a new patch and corresponding installation instructions.*

5.3.1.12 Work unit ALC_PAM.1-11

ALC_PAM.1-11 The evaluator ***shall examine*** if the PAM processes address how patches are securely generated and distributed, including applicable responsibilities and procedures. These processes include:

a) how the user is notified of the availability of a new patch due to a security issue, e.g.:

   — through email;

   — through systematic checks to a website handled by the product.

b) how the patches are made available and securely distributed to the end user, for example:

   — uploaded to a website by the developer and systematically downloaded by the TOE by using an appropriate and declared security protocol;

   — sent to the end-user using delivery services and providing installation instructions where administrator rights shall be implemented using password/authentication codes and/or cryptographic authentication techniques.

ALC_PAM.1.10C *The PMD shall describe all necessary developer procedures to support the patch functionality of the TOE.*

5.3.1.13 Work unit ALC_PAM.1-12

ALC_PAM.1-12 The evaluator ***shall examine*** the implementation of the PMD specified by the developer. For example, implemented procedures for using cryptographic keys or signatures for patches.

If applicable, the evaluator shall examine:

   — How the cryptographic keys involved in signing and/or distributing patches are generated and managed during its entire life-cycle so they have enough strength to protect the authenticity of the updates?

   — How the cryptographic keys are created?

   — How the cryptographic keys are securely stored?

   — How the cryptographic keys used to provide authenticity, integrity, confidentiality or protection against replay or misuse of new patches have a strength commensurate with the evaluation assurance level?

   — How the cryptographic keys are destroyed or archived at the end-of-support of the product?

   — Who approves the releasing of updates?

   — Who can access the cryptographic keys used for signing updates?

ALC_PAM.1.2D *The developer shall provide end-of-support information to the TOE users.*

5.3.1.14 Work unit ALC_PAM.1-13

ALC_PAM.1-13 The evaluator ***shall check*** that end-of-support information is available to the TOE users, e.g. in documents such as the security target (ST), guidance, release notes, and/or information on the product (support) website.

5.3.1.15 Work unit ALC_PAM.1-14

ALC_PAM.1-14 The evaluator ***shall examine*** the end-of-support information to ensure consistency across documents if the information is present in several documents.

5.3.1.16 Work unit ALC_PAM.1-15

ALC_PAM.1-15 The evaluator ***shall check*** that the end-of-support information is unambiguous and complete in the sense that it allows users to determine or put in place the measures to know the date of the end-of-support. For example, end-of-support information can contain:

— end of product maintenance;

— end of product manufacturing;

— end of general availability;

— last order date.

5.3.1.17 Work unit ALC_PAM.1-16

ALC_PAM.1-16 The evaluator ***shall examine*** the end-of-support information of the developer and any corresponding evidence if this gives a rationale for the end-of-support date.

The rationale should allow the end user to consider the end-of-support date into his general TOE risk management.

ALC_PAM.1.4D *The developer shall record evidence of the application of the PMD.*

5.3.1.18 Work unit ALC_PAM.1-17

ALC_PAM.1-17 The evaluator ***shall examine*** evidence of the application of the PMD.

In case the TOE is part of a new product development, evidence from the same developer should be accepted, e.g. evidence from comparable products or product lines.

Alternatively, the developer can execute a dry run of the application of the PMD to generate the necessary evidence.

5.3.1.19 Work unit ALC_PAM.1-18

ALC_PAM.1-18 The evaluator ***shall check*** for results showing the application of the policies.

For example:

— internal policy audit report;

— evidence that the policies have been applied.

5.3.1.20 Work unit ALC_PAM.1-19

ALC_PAM.1-19 The evaluator ***shall check*** if unresolved security issues exist and if these fulfil the policy requirements.

5.3.1.21 Work unit ALC_PAM.1-20

ALC_PAM.1-20 The evaluator ***shall check*** if decisions in the PAM processes have been documented.

5.3.1.22 Work unit ALC_PAM.1-21

ALC_PAM.1-21 The evaluator ***shall check*** the patch release notes for the content required by the PMD.

ALC_PAM.1.5D *The developer shall release patches as defined in the PMD until the end-of-support of the TOE.*

5.3.1.23 Work unit ALC_PAM.1-22

ALC_PAM.1-22 The evaluator ***shall examine*** aspects of the PMD to determine that these are being used.

In addition to examining the procedures themselves, the evaluator seeks some assurance that the procedures are applied in practice, through, for example:

— records of the decisions taken;

— records of the testing done;

— records of self-assessment.

ALC_PAM.1.6D *The developer shall follow the PMD to produce an updated set of evaluation evidence for each released patch at least until the stated end-of-support of the TOE.*

5.3.1.24 Work unit ALC_PAM.1-23

ALC_PAM.1-23 The evaluator ***shall examine*** the implementation of the PMD specified by the developer.

5.3.1.25 Work unit ALC_PAM.1-24

ALC_PAM.1-24 The evaluator ***shall examine*** (updated) evaluation evidence for released patches.

ALC_PAM.1.7D *The developer shall provide a channel used to check for the availability and/or download of patches with means to protect the channel according to the TOE's specified security capabilities.*

5.3.1.26 Work unit ALC_PAM.1-25

ALC_PAM.1-25 The evaluator ***shall examine*** if the required channel for patches is available and provides the security capabilities as specified in the TOE design documentation.

It is important to note that this work unit should be performed in connection with the corresponding work units from ALC_DEL.

6 Additional guidance for evaluators

6.1 General

The following work units list additional activities for evaluators who apply this concept during the initial evaluation of a TOE. The concept assumes a (technical) patch is already available during the evaluation of the initial TOE for the evaluation of the patch mechanism.

If no prefix is given, the text from ISO/IEC 18045 is extended by the words formatted in bold type. If the prefix "add" is given, the evaluators should follow the work unit text in ISO/IEC 18045 and additionally the guidance in this clause presented in bold font. Families and work units that are not listed should not be modified.

The additional activities for evaluators listed in 6.2 to 6.7 shall apply where an assurance component is claimed in the security target.

6.2 Class ASE

6.2.1 ASE_INT

ASE_INT.1-3: The evaluator shall examine the TOE reference to determine that it uniquely identifies the TOE and patches.

6.3 Class ADV

6.3.1 ADV_ARC

ADV_ARC.1-3: (add) **If the patch installation is executed during the (secure) initialisation of the TOE, the security architecture description should contain the details.**

ADV_ARC.1-5: (add) **The evaluator shall examine the security architecture description to determine that it clearly indicates that the patch verification mechanism cannot be bypassed.**

6.3.2 ADV_FSP

ADV_FSP.1-1: (add) **The TSFI should contain interface(s) for the patch installation.**

ADV_FSP.1-2: (add) **The TSFI for patch installation should be SFR-enforcing.**

6.3.3 ADV_IMP

ADV_IMP.1-3: (add) **The sample of the implementation representation should contain a patch example (i.e. test patch).**

6.3.4 ADV_TDS

ADV_TDS.1-1: (add) **The TDS should include a description of patch installation mechanism.**

6.4 Class AGD

6.4.1 AGD_OPE

AGD_OPE.1-1: (add) **The operational user guidance should include descriptions of how the patch installation is executed and any relevant roles.**

AGD_OPE.1-2: (add) **The operational user guidance should include descriptions of the patch installation interfaces.**

6.4.2 AGD_PRE

AGD_PRE.1-1: The evaluator shall examine the provided acceptance procedures to determine that they describe the steps necessary for secure acceptance of the TOE **and patches** in accordance with the developer's delivery procedures.

AGD_PRE.1-2: The evaluator shall examine the provided installation procedures to determine that they describe the steps necessary for secure installation of the TOE **and patches,** and the secure preparation of the operational environment in accordance with the security objectives in the ST.

AGD_PRE.1-3: The evaluator shall perform all user procedures necessary to prepare the TOE **and patches** to determine that the TOE and its operational environment can be prepared securely, using only the supplied preparative procedures.

6.5 Class ALC

6.5.1 ALC_CMC

ALC_CMC.1-1: The evaluator shall check that the TOE **and patches** provided for evaluation are labelled with their references.

ALC_CMC.3-8: The evaluator shall check that the configuration items **including patches** identified in the configuration list are being maintained by the CM system.

6.5.2 ALC_CMS

ALC_CMS.1-1: The evaluator shall check that the configuration list includes the following set of items:

a) the TOE itself **and patches**;

b) the evaluation evidence required by the SARs in the ST.

ALC_CMS.2-1: The evaluator shall check that the configuration list includes the following set of items:

c) the TOE itself **and patches**;

d) the parts that comprise the TOE **and patches**;

e) the evaluation evidence required by the SARs.

ALC_CMS.3-1: The evaluator shall check that the configuration list includes the following set of items:

f) the TOE itself **and patches**;

g) the parts that comprise the TOE **and patches**;

h) the TOE implementation representation **and patches implementation representation**;

i) the evaluation evidence required by the SARs in the ST.

ALC_CMS.5-1: The evaluator shall check that the configuration list includes the following set of items:

j) the TOE itself **and patches**;

k) the parts that comprise the TOE **and patches**;

l) the TOE implementation representation **and patches implementation representation**;

m) the evaluation evidence required by the SARs in the ST;

n) the documentation used to record details of reported security flaws associated with the implementation (e.g. problem status reports derived from a developer's problem database);

o) all tools (incl. test software, if applicable) involved in the development and production of the **TOE and patches** including the names, versions, configurations and roles of each development tool, and related documentation.

6.5.3 ALC_DEL

ALC_DEL.1-1: The evaluator shall examine the delivery documentation to determine that it describes all procedures that are necessary to maintain security when distributing versions of the TOE **and patches** or parts of it to the consumer.

Additionally, the evaluator shall examine delivery related aspects of the PMD specified by the developer. The following questions can be used as guidance.

— How is the update moved from the development environment to the signing environment so that it is not tampered?

— How is the generation of the proof-of-authenticity of new patches carried out in a secure and audited environment, commensurate with the evaluation assurance level?

— How does this process generate logs?

— How are these logs audited?

6.5.4 ALC_DVS

ALC_DVS.1-1: (add) **The documentation of the patch development and deployment environment should be examined as well.**

6.5.5 ALC_FLR

ALC_FLR.1-2: (add) **The evaluator shall examine the root cause analysis for each discovered security flaw, if available.**

6.5.6 ALC_LCD

ALC_LCD.1-1: (add) **The maintenance process should include the patch management (PAM) process.**

The description of the PAM processes should include:

- **Description of the roles and responsibilities inside the organization involved in the patch development.**
- **Patch development responsibilities, e.g. patch development tasks as part of the responsible, accountable, consulted and informed (RACI) matrix, or patch development tasks as a function of a product development team or maintenance team.**
- **Patch release procedures, e.g. procedural steps as part of hardware/firmware/software patch release, quality assurance (QA) test, integration test, or customer release.**
- **Responses to a failure during patch release testing.**

ALC_LCD.1-2: (add) **The evaluator shall select and examine the PAM process life cycle output documentation. A sample of evidence covering each type of relevant event should confirm that all operations of the PMD are carried out in line with the PMD. Types of relevant events are, for example, signing logs, approval of updates, SRR, fulfilled checklists and bug tracker evidence.**

The evaluator may choose to sample the evidence. For guidance on sampling, see ISO/IEC 18045:2022, A.2.

Further confidence in the correct operation of the PMD can be established by means of interviews with selected development staff. Such interviews can complement rather than replace the examination of documentary evidence, but may not be necessary if the documentary evidence alone satisfies the requirement. The evaluator may visit the development site in support of this activity.

The evaluator shall examine aspects of the PMD to determine that these are being used.

In addition to examination of the procedures themselves, the evaluator seeks some assurance that they are applied in practice. One possible approach is a development site visit where practical application of the procedures can be observed (e.g. examine records of the decisions taken, of the testing done, or of self-assessment).

If a site visit is already included in the evaluation plan, the evaluator shall apply this option to check that the processes are applied in practice.

Alternatively, another approach is observing that the process is applied in practice when the evaluator obtains new updates solving the security flaws found during the vulnerability analysis (AVA_VAN).

6.5.7 ALC_TAT

ALC_TAT.1-1: (add) **The evaluator shall check the tools. The list of tools for PAM should include e.g. issue tracking, configuration management and release management.**

6.6 Class ATE

6.6.1 ATE_COV

ATE_COV.1-1: (add) **The test coverage should contain the patch installation interface (i.e. related TSFI).**

6.6.2 ATE_DPT

ATE_DPT.1-1: (add) **The depth of testing analysis should contain the patch installation mechanism (i.e. TSF subsystem).**

6.6.3 ATE_IND

ATE_IND.1-3: (add) **The patch installation mechanism should be part of this test subset, i.e. shall contain at least the installation of a patch example (i.e. test patch).**

6.7 Class AVA

6.7.1 AVA_VAN

AVA_VAN.1-1: The evaluator shall examine the TOE **and patches** to determine that the test configuration is consistent with the configuration under evaluation as specified in the ST.

AVA_VAN.1-3: (add) **To identify potential security flaws in the TOE, the patch installation mechanisms (e.g. used libraries or own implementations) should be analysed.**

AVA_VAN.1-10: The evaluator shall examine the results of all penetration testing to determine that the TOE **and patches**, in the TOE operational environment, are resistant to an attacker possessing a Basic attack potential.

Other work units from AVA_VAN should be applied accordingly.

Annex A
(informative)

Options for evaluation authorities

A.1 General

This annex outlines several options for evaluation authorities aiming to use ALC_PAM.1 to establish trust models between the parties, i.e. the developer, the evaluation facility and the evaluation authority, which can facilitate the assurance maintenance process.

Although certification aspects are not in the scope of the ISO/IEC 15408 series, this annex uses the terms "(product) certification" and "(product) certificate" to refer to the activity of an evaluation authority regarding an evaluated product and to the result of such activity, respectively.

A.2 Assumptions

The following assumptions should be met in order to use the options given in this annex:

a) Options are only available if the same pair of evaluation authority/IT security evaluation facility (ITSEF) runs the activities for a patch re-evaluation.

b) In case security flaws were identified in an initial or previous TOE, a root cause analysis should be completed by the developer, ITSEF and evaluation authority.

A.3 Option 1: Provide a fast-track certification process

Developers that implement the TOE security objectives (with corresponding SFR from this document or equivalent) and operational environment security objectives which are aligned with the requirements from ALC_PAM, can be allowed to access fast-track certification processes by evaluation authorities. Those fast-track certification processes can be limited to security flaws. Evaluation authorities can create a fast-track priority queue for processing these certifications.

Developers are still required to evaluate the changes with an ITSEF under the evaluation authority, but this evaluation can start without previous authorization by the evaluation authority.

Furthermore, security flaw patches are typically attached to a patch with other updates. Unless there is a major security fix, most vendors do not issue an out-of-cycle patch and instead include multiple changes (beyond simple security fixes) in patches/updates released. In this case, all changes shall be identified and reviewed for impact. The fast tracking is possible, if the patches only contain security fixes, thus making it feasible to speed up the process.

Changes in the hardware of the TOE, the hardware of the operational environment or in the documentation can be other reasons to initiate a new evaluation and facilitate this with a fast-track process.

A.4 Option 2: Define different types of updates and associated certification processes

Different types of updates can be defined for IT products to support associated certification processes.

Some evaluation schemes or recognition agreements have defined, for example, major and minor as types of updates. What is covered by such types of updates is subject to the evaluation schemes and is beyond the scope of this document.

The following aspects should be considered for the definition of types of updates:

— need for updated assurance evidence compared to the initial evaluation, e.g. changes that affect the TOE, or only non-TOE parts of the product, or only the TOE environment;

— changes in the design or the (security) architecture of the TOE;

— changes in the source code of the TOE, including number and amount of changes;

— correction of one or multiple security flaws;

— correction of one or multiple functional flaws, but no functional enhancements;

— functional changes or new functionality.

The evaluation authority can define criteria for such aspects and can assign different types of updates to these.

A.5 Option 3: Support re-use of evaluation results

Developers who claim ALC_PAM are able to immediately provide evidence to future re-evaluations.

For example, the Common Criteria Recognition Arrangement already allows to re-use evaluation results. But compared to existing practices, ALC_PAM encourages developers to continuously generate this evidence during product maintenance.

To support fast and plannable re-evaluation, the evaluation authority can also publish scheme policies that describe how and which evaluation results can be re-used in future re-evaluations. The combination of fresh evidence from latest patch development and re-use of previous evaluation results can support an efficient certification process.

A.6 Option 4: Re-evaluation performed by the same evaluator

If the re-evaluation of the TOE is performed by the same ITSEF and even by the same evaluator or evaluation team, the requirements for the re-evaluation may be adjusted. For example, the ITSEF can decrease the reporting requirements or the acceptance procedure of the evaluation reports can be accelerated.

A.7 Option 5: The non-certified ETR-based approach

If a patch fixes a security flaw (known or not), there is a need for the developer:

— to update ATE so that the absence of the flaw is demonstrated and documented;

— possibly to update other parts of the TOE documentation, so as to clarify why the flaw was not discovered by the developer nor the lab during the first evaluation.

There is also a need for the evaluator:

— to review the developer evidence;

— to independently assess whether the security flaw was correctly analysed and fixed by the developer;

— possibly to check for the existence of similar errors elsewhere in the TOE;

— possibly to update their AVA_VAN analysis, so as to clarify where the flaw was not discovered by the evaluator during the first evaluation.

In this approach, the user of the TOE can obtain assurance information from the evaluation technical report (ETR). The evaluation authority does not provide direct oversight of this process.

A precondition of this option is that the user of the TOE trusts the ITSEFs and decides to rely on ETRs delivered by ITSEFs (without the evaluation authority overview) in in-between re-evaluations. To help this, evaluation authorities who follow this option support their licensed ITSEFs so that they are technically and methodologically proficient, to minimize the risks of errors in the non-validated ETRs produced in in-between re-evaluations.

The feasibility of this option highly depends on the policies of the evaluation authorities and expectations users (or risk owners) of the TOE.

A.8 Option 6: Provide templates to analyse the impact of changes of a patch

This document also provides a template as a starting point for evaluation authorities in Annex B.

A.9 Option 7: Continued trust in products that have been certified against patch management criteria

Updates addressing security flaws can be accepted by default because of the additional assurance resulting from ALC_PAM. The patched version can be considered under the maintenance report just with the ITSEF criteria.

A.10 Option 8: Penalties if developers do not follow the published rules

As part of the certificate monitoring, the evaluation authority can apply penalties, e.g. suspension of the certificate.

Penalties can be applied if developers do not submit a patched product for re-evaluation in a defined time frame, or if developers provide incorrect evidence to the ITSEF.

If a fast-track certification process is available, developers can be denied access to this if they do not follow the published rules.

A.11 Option 9: Mandate root cause analysis by the ITSEF

While it is assumed that ISO/IEC 15408 can provide a high level of assurance, this does not imply that products are 100 % free of bugs. This can be due to:

— Security flaws that were not exploitable in the evaluated operational environment.

— Security flaws that fallen out of the applicable attack potential.

— When protection profiles providing test cases are used, it is possible that these test cases have been performed incorrectly.

— Use of sampling procedures.

— Problems arising from the processes and flaw analysis methodologies of the lab.

The presence of security flaws in an evaluated TOE should always require a root cause analysis to investigate why it was not discovered by the ITSEF and to avoid new similar problems in the affected TOE and other TOEs evaluated by the same ITSEF.

Annex B
(informative)

Template for the security relevance report

Table B.1 gives a template for the security relevance report (SRR) defined in this document. The SRR describes the security relevance of the planned patch. The planned patch can deal with one or more flaws or issues.

Developers can adjust the template based on given circumstances.

Table B.1 — Template for the security relevance report

Flaw or issue	Description	Options for mitigation	Related change	Security impact
Includes reference to the flaw or issue	Security relevance consideration, e.g. remote code execution, or only product type specific flaw. Category criteria: e.g. common weakness enumeration (CWE)	e.g. change product/TOE, new guideline (special configuration)	Relation to the configuration management (CM)	e.g. security bug-fix, functional correction, new feature

Annex C
(informative)

ALC_PAM PMD examples

C.1 General

The patch management of CC product/TOE developers shall have the content of the patch management documentation (PMD) as defined in ALC_PAM.1. This annex gives an example of an outline of such a (PMD) policy.

The policy should include these aspects:

a) Monitoring of flaws and issues

b) SRR result categories

c) Assessment of flaws and issues, or Patch integration (or change) criteria

d) Policies to maintain CC/ALC development process

e) Policies for patch releases

f) Updated guidance

g) Self-assessment and confirmation of the application of existing policies on a regular basis

C.2 Monitoring of flaws and issues

Developers should monitor multiple sources for information on flaws and issues. All security relevant flaws and issues shall be analysed by the developer. The result shall be documented in the SRR report.

The roles and responsibilities for gathering the information and the initial flaw and issue assessment shall to be defined.

The following are examples of flaw and issue sources which are monitored:

— security@company E-Mail inbox

— internally detected flaws, e.g. by QA team

— flaws and issues reported by customers

— third party library related flaws, e.g. open source libraries

The product security officer is responsible for monitoring incoming candidate flaws and issues.

C.3 SRR result categories

At least two categories shall be defined, i.e. a first category whereby no patch is required, a second category whereby patch is required.

Developers are encouraged to define the categories which describe their business perspective, i.e. specific policies based on customer contracts or based on requirements for regulated use-cases.

In the following example, the definition for the two types of categories is given:

— Category 1 “internal QA”: e.g. functional corrections not affecting the TSF, security bugfixes that do not require an update of the ADV evidence. If the whole patch has been qualified for this category, the testing of the patch is done by the QA team.

— Category 2 “re-evaluation”: e.g. functional correction or security update of the TSF which requires for updates of the ADV evidence. If at least one change is qualified for this category, the developer starts the re-evaluation immediately.

C.4 Assessment of flaws and issues

For the product (or TOE) lifetime, the developer shall define their internal criteria to assess flaws and issues.

The criteria shall to be used to decide if the flaw remediation will be one of the following types:

— technical correction, i.e. release of a patch,

— publication of additional guidance, i.e. configuration or procedural workaround, or

— recommendation to change the product setup, e.g. the installation of technical compensating countermeasures (e.g. additional firewall packet filter).

The developer is able to handle multiple flaws by clustering the required changes into one single patch.

The handling of the flaws shall be documented as part of the SRR.

For example, the developer defines a policy that uses the following criteria:

— complexity of backports;

— operational stability (development teams are able to estimate effect for operational stability);

— security impact;

— customer impact (i.e. practical problems, theoretical problems)

— timely impact, (i.e. customers expect patches each quarter of a year, or timely resolution of minor security problems);

— third-party library related flaws and issues:

  — update only libraries that are still supported as well,

  — backport latest changes to used library version, or

  — upgrade to latest library version.

The product security officer is responsible for the assessment of incoming candidate flaws and issues.

C.5 Policies to maintain CC/ALC_PAM process

The developer defines how the CC/ALC_PAM process is maintained during the product (TOE) lifetime.

NOTE The baseline evaluation has shown the developer’s capability to develop and produce a product according to the CC requirements. This policy aspect requires the developer to setup maintenance procedures, showing how all CC/ALC_PAM evidence is generated in parallel to the default product (TOE) maintenance.

EXAMPLE The product security officer is responsible for maintaining the evaluation input like design documents (ADV). The QA team is responsible for re-running the developer tests (ATE_FUN).

C.6 Policies for patch releases

The policies below shall be followed before the next patch is released:

— definition of internal release stages and policies;

— process definition with failure/cancel criteria for validation tests and follow-up procedures for these cases;

— definition of cases if the external evaluation facility should be contacted, and if additional tests should be performed before patch release;

NOTE These cases do not directly address certificate updates but are related to the involvement of the evaluation facility without (full) re-evaluation.

— definition of ruleset for roles (e.g. development, QA department, product owner) in the patch release process;

— responsible role for the final patch release decision;

— unique label for each patch to identify all release items.

The policies can differentiate between the different SRR result categories.

C.7 Updated guidance

For each patch released, the developer shall verify if a guidance update is required. The details shall be defined in a policy. The following reasons can be considered for the policy definition:

— exceptions to let flaws or issues unhandled but guidance how to mitigate these flaws, e.g. with procedural changes;

— update or installation pre-conditions, e.g. hardware requirements should be documented.

C.8 Self-assessment and confirmation of the application of these policies

The developer shows periodically that the policies are applied. This should be shown by (partly) published results of the self-assessment.

The commitment of the developer shall be documented as part of the policy.

EXAMPLE The summary of the results of the annual self-assessment is published on the developer's website with reference to the related product certification IDs. The self-assessment is supported by an external audit team leader to ensure independence from the development team's perspective.

Annex D
(informative)

Patch management functional package example

D.1 General

This annex includes an example of a functional package, showing how to write a patch management security problem definition (SPD), corresponding objectives for a TOE and security functional requirements (SFRs).

D.2 Security problem definition

D.2.1 General

This SPD addresses local and remote attacks that are relevant in the context of patch installation.

This annex includes two options of how regular checks for patches should be realized. Option A considers patch checking is a functionality of the TOE. Option B requires this activity to be realized by the operational environment of the TOE.

NOTE This annex does not cover all relevant aspects for secure patch management. For example, security of the key infrastructure and secure storage of keys are not addressed.

D.2.2 Assumptions

The SPD includes the following assumption:

a) **A.PAM.RESPONSIBLE_USERS:** Users responsible for patching put adequate measures to receive the patch notifications and allow the loading, installation, and activation of the patches. The responsible users support any activity which is required to perform the patching process, including the availability of the direct or indirect communication channel between the patch issuer and the loader.

D.2.3 Threats

The SPD includes the following threats:

a) **T.PAM.INSECURE_TOE:** An attacker blocks the ability of the TOE to get new security patches, preventing the user from updating it. Future detected security flaws of the TOE will not be corrected despite the availability of a new security patch.

b) **T.PAM.ROGUE_PATCH:** An attacker forges a rogue malicious patch, which is indistinguishable from a legitimate patch or able to violate the integrity of the patch mechanism. The rogue malicious patch is installed or processed by the TOE, altering the intended TSF functionality.

c) **T.PAM.INSECURE_LOAD:** An attacker can subvert the TOE to allow loading a patch by an unauthorized entity and/or to load an authorized patch that breaks the TOE patching policy.

D.2.4 Organizational security policies

The SPD includes the following organizational security policy (OSP):

a) **OSP.PAM.PATCH_CHECKING:** Users in the operational environment of the TOE regularly check for new patches.

D.3 Objectives

D.3.1 General

The objectives are composed of operational environment security objectives and TOE security objectives.

D.3.2 Operational environment security objectives

Operational environment security objectives include:

a) **OE.PAM.NOTIFICATION:** Users responsible for patching shall put adequate measures to receive the patch notifications from the patch issuer.

b) **OE.PAM.PATCH_ACTIVATION**: The responsible users shall allow the loading, installation, and activation of the patches.

c) **OE.PAM.PATCH_SUPPORT:** The responsible users for patching shall support any activity which is required to perform the patching process, including the availability of the direct or indirect communication channel between the patch issuer and the Loader.

d) **(option B) OE.PAM.PATCH_CHECKING:** Users responsible for patching shall use or provide a communication channel and regularly check for new security patches and notify TOE administrators of the availability of the updates according to a defined policy.

ST/PP author shall select between implementing patch checking in the TOE (option A) or in the operational environment (option B).

D.3.3 TOE security objectives

TOE security objectives include:

a) **(option A) O.PAM.PATCH_CHECKING:** The TOE shall regularly check for new security patches and notify TOE administrators of the availability of the updates according to a defined policy.

ST/PP author shall select between implementing patch checking in the TOE (option A) or in the operational environment (option B).

b) **O.PAM.TRANSPORT_SECURITY:** The channel used to check for the availability of patch(s) and/or download of patch(s) shall be protected in the security dimensions defined.

c) **O.PAM.SECURE_LOAD:** The loader shall check the authenticity of the entity trying to load the patch. The Loader shall enforce the patching policy to ensure only authorized patches are loaded.

Application note: The patching policy can describe constraints for the patch loading from a TOE perspective (e.g. version rollback is prevented by the TOE) or an organizational perspective (e.g. checking of hardware constraints before installation of the TOE, only allow installation of patching between certain hours of the day).

d) **O.PAM.ACTIVATION:** Activation of the patch and update of the identification data shall be performed as an atomic operation. All the operations needed for the code to be able to operate as in the final TOE shall be completed before activation. If the activation is successful, then the resulting product is the final TOE.

e) **O.PAM.ERROR**: In case of interruption or incident which prevents the forming of the final TOE (i.e tearing, integrity violation, error case…), the initial TOE shall remain in its initial state or fail secure. i.e. it may be restored.

D.3.4 TOE security objective rationale

The mapping of the threats, assumptions and OSPs to the objectives and objectives of the environment is given in Table D.1.

Table D.1 — Security objectives rationale

	O.PAM. PATCH_ CHECKING	O.PAM. TRANSPORT_ SECURITY	O.PAM.SE-CURE_LOAD	O.PAM.ACTI-VATION	O.PAM.ERROR	OE.PAM. NO-TIFI CATION	OE.PAM. PATCH_ACTI-VATION	OE.PAM. PATCH_SUP-PORT	OE.PAM. PATCH _CHECKING
T.PAM.INSECURE_TOE	**A**	**X**				**X**	**X**		**B**
T.PAM.ROGUE_PATCH		**X**	**X**	**X**	**X**				
T.PAM.INSECURE_LOAD			**X**						
A.PAM.RESPONSIBLE_USERS						**X**	**X**	**X**	
OSP.PAM.PATCH_CHECKING									**X**

T.PAM.INSECURE_TOE: This threat is mitigated by the operational environment **OE.PAM. NOTIFICATION** which will provide means to notify of the availability of new security patches to end users. The responsible users of the TOE will support the activation of available patches (**OE.PAM. PATCH_SUPPORT**).

If **O.PAM.PATCH_CHECKING** (option A) is implemented by the TOE, the TOE will check systematically for new updates, using a protected channel (**O.PAM.TRANSPORT_SECURITY**).

Otherwise, this functionality will be provided by the operational environment through **OE.PAM. PATCH_CHECKING** (option B).

T.PAM.ROGUE_PATCH: This threat is mitigated by the joint force of security objectives for the operational environment and security objectives for the TOE.

The TOE itself have mechanisms to verify the entity trying to load the patch (**O.PAM.SECURE_LOAD**). Only after successful verification of the signature, the TOE processes and installs the patch in an atomic way (**O.PAM.ACTIVATION**) so no dangerous TSF mediated actions are allowed. In case of an error, **O. PAM.ERROR** will prevent the operation of the TOE in a failure state, restoring the TOE to its initial state.

When the update is downloaded from an update provider, this communication will be protected by **O.PAM.TRANSPORT_SECURITY.**

T.PAM.INSECURE_LOAD: The loader enforces that the entity loading the patches is authorized (**O.PAM. SECURE_LOAD**). Additionally, the loader enforces that patches are only loaded according to a defined patching policy (**O.PAM.SECURE_LOAD**). This policy can include statements such as the requirement for an authenticated administrator to install a patch, the prohibition to install older versions of the TOE, or requirements compliant with the underlying platform.

A.PAM.RESPONSIBLE_USERS: This assumption is upheld by the combination of **OE.PAM. NOTIFICATION**, **OE.PAM.PATCH_SUPPORT** and **OE.PAM.PATCH_ACTIVATION**.

OSP.PAM.PATCH_CHECKING: This organizational security policy is demanded directly by OE.PAM. PATCH_CHECKING.

D.4 Relationship with JIL supporting documents

In Table D.2, the objectives listed in D.3.3 are compared to the joint interpretation library (JIL) objectives[3] (or ANSSI-CC-NOTE-06/2.0[4]). Table D.2 shows how the objectives can be mapped.

Table D.2 — JIL and TOE security objectives comparison

D.3.3	JIL	Differences/notes
O.PAM.SECURE_LOAD The loader shall check the authenticity of the entity trying to load the patch. The loader shall enforce the patching policy to ensure only authorized patches are loaded.	**"O.Secure_Load_ACode** Secure loading of the Additional Code The Loader of the Initial TOE shall check an evidence of authenticity and integrity of the loaded Additional Code. The Loader enforces that only the allowed version of the Additional Code can be loaded on the Initial TOE. The Loader shall forbid the loading of an Additional Code not intended to be assembled with the Initial TOE. During the Load Phase of an Additional Code, the TOE shall remain secure." [3]	
O.PAM.ACTIVATION Activation of the patch and update of the identification data shall be performed as an atomic operation. All the operations required for the code to be able to operate as in the Final TOE shall be completed before activation. If the Activation is successful, then the resulting product is the Final TOE. **O.PAM.ERROR** In case of an interruption or incident which prevents the forming of the final TOE (i.e. tearing, integrity violation, error case…), the initial TOE shall remain in its initial state or fail secure. i.e. can be restored.	**"O.Secure_AC_Activation** Secure activation of the Additional Code Activation of the Additional Code and update of the Identification Data shall be performed at the same time in an Atomic way. All the operations needed for the code to be able to operate as in the Final TOE shall be completed before activation. If the Atomic Activation is successful, then the resulting product is the Final TOE, otherwise (in case of interruption or incident which prevents the forming of the Final TOE such as tearing, integrity violation, error case…), the Initial TOE shall remain in its initial state or fail secure." [3]	
None	**"O.TOE_Identification** Secure identification of the TOE by the user The Identification Data identifies the Initial TOE and Additional Code. The TOE provides means to store Identification Data in its non-volatile memory and guarantees the integrity of these data. After Atomic Activation of the Additional Code, the Identification Data of the Final TOE allows identifications of Initial TOE and Additional Code. The user shall be able to uniquely identify Initial TOE and Additional Code(s) which are embedded in the Final TOE." [3]	This document allows users to, for example, fully replace a software TOE so there is no distinction between the version of the additional code and the version of the Initial TOE. When the atomic activation is performed, the identification data may change from the version of the initial TOE to the version of the final TOE or to the version of the initial TOE + installed patch(es).

Table D.2 *(continued)*

D.3.3	JIL	Differences/notes
O.PAM.PATCH_CHECKING The TOE shall regularly check for new security patches and notify TOE administrators of the availability of the updates according to a defined policy.	None	This new security objective requires the TOE to systematically check for updates according to a defined policy (which can be empty). This will allow final users to stay aware of new patches.
O.PAM.TRANSPORT_SECURITY The channel used to check for the availability of patch(s) and/or download of patch(s) shall be protected in the security dimensions defined.	None	This new security objective requires the TOE to be able to protect the channel used to download new patch(es) in the security dimensions defined in a policy (which again can be empty). This enables protection of confidentiality/integrity of the patches during transport.

D.5 How to write/select security functional requirements

In light of the different TOE-types and different security needs for patch functionality, this document does not specify one set of security functional requirements (SFRs) for patch management functionality. This clause gives guidance on different ways of writing SFRs. In addition, D.7 provides an example for a set of SFRs describing patch management functionality.

This clause describes how a secure patching functionality can be modelled using only part two components.

The model is based on the use of two policies, the first one to control the information flow from the entity providing updates to the TOE, and the second one to control the access of the TSF to the update in order to perform a secure installation.

Both policies use the subject S.Loader to describe the part of the TSF that performs this actions. S.Loader has a set of security attributes, providing a high degree of flexibility, and allowing the TOE to be highly configurable in regards to its defined security attributes, so it is expected that the TSS describes what is configurable and to what extent. In case something is not configurable, the applicable values shall be precisely defined (e.g. if the policy for patch checking is not configurable, the hardcoded policy shall be described).

The information flow policy guarantees that the patch is adequately downloaded using the means selected by the ST author to protect the channel. These means can include physical protection, the use of cryptographic functionalities or other applicable SFRs like trusted channels. Those SFRs shall be mapped to O.PAM.TRANSPORT_SECURITY and O.PAM.PATCH_CHECKING.

This information flow can be automatically exercised in a defined way potentially notifying the end user of the availability of the patch, if needed.

When a patch has been downloaded, the access control policy guarantees that it is only installed when a cryptographic check has been performed to verify the authenticity and integrity of the update and providing, if needed, other security characteristics such as confidentiality.

This same access control policy also allows configuration of the security attributes of the subject S.Loader.

The final import of the patch into the TOE is only allowed by means of activation and it is guaranteed that in case of error, the TOE remains in a secure state.