Chapter 1 Introduction

This section identifies the PP module as well as the base PP and provides a module overview for potential users.

1.1 PP Module Reference

Title: MILS Platform Protection Profile Secure Update Module
Sponsor: certMILS Consortium
CC Version: 3.1 (Revision 5)
Assurance Level: see the Base PP.
Version: draft
Keywords: Base-PP, PP module, Operating System, Separation Kernel, MILS

1.2 Base PP Identification

Base MILS Platform Protection Profile, Version: 1.0

1.3 PP Module Overview

This module supplements the base PP by specifying the minimum functionality for a secure update process that a TOE has to provide.

Chapter 2 Consistency Rationale

This section states the correspondence between the PP module and its base PP.

2.1 TOE Type Consistency

The TOE type for which both the base PP and this PP module are designed is “a special kind of operating system, namely an SK.”

An SK is a special kind of operating system that allows to effectively separate different containers called “partitions” from each other. Applications themselves are hosted in those partitions. They can also be entire operating systems. The SK is installed and runs on a hardware platform (e.g. embedded systems, desktop class hardware).

The PP module extends the base PP by specifying security objectives covering functions relative to a secure update process.

2.2 Security Problem Definition Consistency

2.2.1 Assets

The base PP describes the assets to be protected:

- Memory (AS.MEM)
- CPU time (AS.TIME)

This PP module does not add any asset.

2.2.2 Threats

The base PP describes the threats contemplated:

- T.DISCLOSURE
- T.MODIFICATION
- T.DEPLETION

This PP module contemplates the following additional threats:

- T.UNAUTHORIZED_UPDATE
- T.MANIPULATED_UPDATE
- T.TSF_NOT_OPERATIONAL

The threats T.UNAUTHORIZED_UPDATE and T.MANIPULATED_UPDATE bring specific compatible scenarios associated to T.MODIFICATION.

The threat T.TSF_NOT_OPERATIONAL brings a specific compatible scenario associated to T.DEPLETION.

2.2.3 Organizational Security Policies

Neither the base PP nor this PP module define organizational security policies.

2.2.4 Assumptions

This PP module defines the following additional assumptions:

- A.SECURE_AUTHENTICATOR

This additional assumption is compatible with the assumptions defined in the base PP. The assumptions included in the base PP are applicable with no changes.

2.3 Security Objectives Consistency

The base PP describes the security objectives to be implemented:

- OT.CONFIDENTIALITY
- OT.INTEGRITY
- OT. AVAILABILITY

This PP module adds the following security objectives for the TOE:

- OT.AUTHENTICATE_ORIGIN
- OT.DETECT_MANIPULATION
- OT.OPERATIONAL_TSF

These security objectives add security functionality to the TOE regarding the secure update process which is compatible to the rest of security objectives for the TOE defined in the base PP.

2.4 Security Functional Requirements Consistency

In addition to the set of SFRs included in the base PP, this PP module defines:

- FDP_DAU.1 Basic Data Authentication – This SFR is compatible with the set of SFRs defined in the base PP as it adds independent and specific functionality for authenticating the data used for the secure update process.
- FPT_TIM.1 TSF Integrity Monitoring and Action – This SFR is compatible with the set of SFRs defined in the base PP as it adds independent and specific functionality for checking the integrity and completeness of the data used for the secure update process.
- FDP_ROL.1 Basic Rollback – This SFR is compatible with the set of SFRs defined in the base PP as it adds independent and specific functionality allowing a rollback functionality when errors occurs during the update process.
- FMT_SMF.1 Specification of Management Functions – This SFR is compatible with the set of SFRs defined in the base PP, as it extends the SFR FMT_SMF.1 of the base PP with additional management functionality.
- FMT_MTD.1 Management of TSF Data – This SFR is compatible with the set of SFRs defined in the base PP as it adds independent and specific management functionality associated to the data used for the authenticity and integrity verification of the updates.

Chapter 3 Conformance Claim

This protection profile module claims conformance to

- Common Criteria for Information Technology Security Evaluation. Part 1: Introduction and general model. Version 3.1, Revision 5. April 2017. CCMB-2017-04-001 [1]
- Common Criteria for Information Technology Security Evaluation. Part 2: Security Functional Components. Version 3.1, Revision 5. April 2017. CCMB-2017-04-002 [2]
- Common Criteria for Information Technology Security Evaluation. Part 3: Security Assurance Components. Version 3.1, Revision 5. April 2017. CCMB-2017-04-003 [3]

as follows

- Part 2 extended.

The “Common Methodology for Information Technology Security Evaluation, Evaluation Methodology; Version 3.1, Revision 5, April 2017. CCMB-2017-04-004, [4]” has to be taken into account.

This protection profile module is associated with the Base MILS Platform Protection Profile Version 1.0.

3.1 Conformance Rationale

Since a PP module cannot claim conformance to any protection profile, this section is not applicable.

3.2 Conformance Statement

This PP module requires strict conformance of any ST or PP claiming conformance to this PP module.

Note: claiming conformance to this PP module also requires claiming conformance to the Base MILS Platform Protection Profile.

Chapter 4 Security Problem Definition

This section describes the security aspects of the environment in which the TOE claiming conformance with the PP module will be used and the manner in which the TOE is expected to be employed. It provides the statement of the TOE security environment, which identifies and explains all:

- Known and presumed threats countered by either the TOE or by the security environment.
- Organizational security policies with which the TOE must comply.
- Assumptions about the secure usage of the TOE.

4.1 Threats

The threat countered by this PP module is the threat of loading a new version of TSF software and data during operation that is not authenticated or has been altered in an unauthorized way, thereby allowing an attacker to compromise the security of the system completely.

Threat Agent

The threat agent is any untrusted subject that has the ability to insert untrusted code or data during an update process of TSF software/data.

Threats agents are therefore subjects that attempt to insert unauthorized software or data during the process of updating TSF software/data. This can be done for example by either initiating an update process without being authorized to do so or by intercepting and manipulating software and/or data that is transmitted to the TOE as part of the update process.

T.UNAUTHORIZED_UPDATE

An attacker may initiate an update process for TSF software/data and thereby insert untrusted code into the TSF without the TSF being able to detect this.

T.MANIPULATED_UPDATE

An attacker may intercept and manipulate software or data transmitted from another trusted IT product to the TOE as part of the update process thereby inserting untrusted code or data into the TSF.

T.TSF_NOT_OPERATIONAL

An attacker may attempt to tamper with the update process to bring the TSF into a state where it is no longer operational.

4.2 Organizational Security Policies

This module defines no organizational security policies.

4.3 Assumptions

A.SECURE_AUTHENTICATOR

The TSF is in the possession of data it can use to authenticate the authenticity and integrity of data transmitted from a trusted entity.

Chapter 5 Security Objectives

Security objectives are concise, abstract statements of the intended solution to the problem defined by the security problem definition (see previous section). The set of security objectives for a TOE form a high-level solution to the security problem. This high-level solution is divided into two part-wise solutions: the security objectives for the TOE, and the security objectives for the TOE's operational environment.

This section presents the solution to the security problem in terms of objectives for the TOE and its operational environment.

5.1 Security Objectives for the TOE

OT.AUTHENTICATE_ORIGIN

A TOE compliant with this PP module provides a mechanism that allows to verify the origin of software and data it receives as part of the update process for TSF code or data.

OT.DETECT_MANIPULATION

A TOE compliant with this PP module provides a mechanism that allows to detect any unauthorized modification of software and data it receives as part of the update process for TSF code or data.

OT.OPERATIONAL_TSF

A TOE compliant with this PP module provides a mechanism that allows the TSF to roll back to its old and verified version in the case the update process fails or is interrupted.

5.2 Security Objectives for the Operational Environment

OE.SECURE_AUTHENTICATOR

The data the TSF holds for authenticating data received as part of a secure update process and verifying its integrity. The initial data for this process has been installed within the TOE in a trusted way.

5.3 Security Objectives Rationale

A secure update process has to ensure that any software and/or data loaded as part of this process is authenticated and integrity verified before it is accepted as an update.

Sometimes the authentication and integrity verification is based on cryptographic mechanisms that require some information like keys to be kept confidential. This information then has to be stored in a location that the assumed threat agent cannot read or modify without violating the assumption of a physically protected environment.

Chapter 6 Extended Components Definition

This section includes possible extended functional components definitions where new functional components not included in CC Part 2 are introduced.

6.1 FPT_TIM TSF integrity monitoring

6.1.1 Family Behaviour

FPT_TIM.1 is identical to FDP_SDI.2 defined in the CC except that it applies to TSF and TSF data.

6.1.2 Component Levelling

The FPT_TIM family contains only one component: FPT_TIM.1.

FPT_TIM.1 is, therefore, not hierarchical to any other component within the FPT_TIM family.

6.1.3 Management

See management description specified for FDP_SDI.2 in [CC].

6.1.4 Audit

See audit requirement specified for FDP_SDI.2 in [CC].

6.1.5 FPT_TIM.1 TSF Integrity Monitoring and Action

Hierarchical to: No other component

Dependencies: No dependencies

FPT_TIM.1.1 The TSF shall monitor [selection: TSF code, TSF data, [assignment: parts of TSF code, parts of TSF data]] for [assignment: integrity errors] using the following rules: [assignment: rules that define how the integrity is verified].

FPT_TIM.1.2 Upon detection of a data integrity error, the TSF shall [assignment: action to be taken].

6.1.6 Rationale

[CC] defines integrity verification for user data. This SFR extends this functional claim to TSF data and TSF code.

Chapter 7 Security Requirements

This section defines the Security Functional requirements (SFRs). This PP module does not introduce specific assurance requirements. The assurance requirements are defined by the Base MILS Platform Protection Profile.

7.1 Security Functional Requirements

7.1.1 Mandatory SFR: Data Authentication

7.1.1.1 FDP_DAU.1 Basic Data Authentication

FDP_DAU.1.1 The TSF shall provide a capability to generate evidence that can be used as a guarantee of the validity of [data received as part of the update process of the TSF and ensure that it comes from a source trusted for providing such information].

FDP_DAU.1.2 The TSF shall provide [the TSF] with the ability to verify evidence of the validity of the indicated information.

Application Note: The mechanism used for implementing this SFR must ensure that the TSF is able to verify that the data receives as part of the TSF update process comes from a source it trusts to provide such data. The mechanism used must also be able to verify that the data received is intended to be part of an update process and contains sufficient information to allow the TSF to identify which parts of the TSF are intended to be updated and how the data received has to be applied in the update process.

7.1.2 Mandatory SFR: Protection of the TSF

7.1.2.1 FPT_TIM.1 TSF Integrity Monitoring and Action

FPT_TIM.1.1 The TSF shall monitor *code and data it receives as update or extension to TSF code or data* for [unauthorized modification of that TSF code and TSF data] using the following rules:

[selection:

- cryptographic keyed message digest compliant with [assignment: keyed message algorithm according to a defined standard] calculated over the TSF and TSF data to be loaded;
- digital signature using [assignment: digital signature algorithm according to a defined standard] of the TSF and TSF data;
- [assignment: other Integrity verification mechanism]
-].

Application Note: If a replay of an old version of update data also needs to be detected, the mechanism to do so needs to be defined in the assignment on ‘other integrity verification mechanism’.

Application Note: if a cryptographic function is used for integrity verification this function also has to be defined as a SFR within the Security Target including the SFRs that define how cryptographic keys are generated or imported and the SFRs defining the protection of those keys.

FPT_TIM.1.2 Upon detection of a data integrity error including the detection of incomplete data, the TSF shall *reject the data received and not perform the update*.

7.1.3 Mandatory SFR: Rollback

7.1.3.1 FDP_ROL.1 Basic Rollback

FDP_ROL.1.1 The TSF shall enforce [no access control policy] to permit the rollback of the [update process of the TSF and restore it as it was before starting the update] on the [detection of an incomplete update and [assignment:: list of other errors that can occur during the update process]].

FDP_ROL.1.2 The TSF shall permit operations to be rolled back within the [boundary of a defined version and state of the TSF].

7.1.4 Optional management SFRs in case the integrity authentication and/or integrity verification data can be updated

7.1.4.1 FMT_SMF.1 Specification of Management Functions

FMT_SMF.1.1 The TSF shall be capable of performing the following management functions: [updating the data that is used to verify the integrity and authenticity of the data received for update].

7.1.4.2 FMT_MTD.1 Management of TSF Data

FMT_MTD.1.1 The TSF shall restrict the ability to *update or add* the [data used to verify the authenticity and integrity of TSF updates] to [a dedicated process within the TSF that requires the following conditions to be satisfied before starting this process [assignment: conditions that must be met to start this process] and uses the following method to verify this data [assignment:: method used to verify the authenticity and integrity of the new data]].

Application Note: If the data used for the verification of the authenticity and integrity of a general TSF update can be managed by either replacing the existing data or by adding new data that can be used for that purpose, this SFR shall describe how this is done in a secure and authenticated way. The process to update this data may be different from the general TSF update process and therefore it is useful to have a separate SFR for this function. Note that if the confidentiality of this data needs to be guaranteed, the conditions that must be met need to describe how the data is kept confidential (e. g. by being transmitted over a secure channel, which then requires an additional SFR for the secure channel).

7.2 Security Requirements Rationale

FPT_DAU.1 addresses the security objective OT.AUTHENTICATE_ORIGIN by requiring that the data received is valid and comes from a source that is trusted to provide such data.

FPT_TIM.1 addressed the security objective OT.DETECT_MANIPULATION by requiring that the TSF is able to detect modifications for the code and data loaded as part of the update process.

FDP_ROL.1 addresses the objective OT.OPERATIONAL_TSF by requiring that the TSF has a rollback process that rolls back to a valid version of the TSF in the case the update process fails or is interrupted.

The optional SFRs ensure that the objectives OT.AUTHENTICATE_ORIGIN and OT.DETECT_MODIFICATION are also met when the data used to verify the integrity and authenticity of the data received for the general update process can themselves be updated. SFRs FMT_SMF.1 and FMT_MTD.1 ensure that such an update can only be performed when the required conditions are met and an update process is used that itself is secure.

7.3 Security Functional Requirements Dependencies Analysis

The following dependencies are defined for the SFRs used in this PP module:

SFR	Dependencies	Satisfied?

FPT_DAU.1	None	Yes
FPT_TIM.1	None	yes
FDP_ROL.1	FDP_ACC.1 or FDP_IFC.1	no
FMT_SMF.1	none	yes
FMT_MTD.1	FMT_SMR.1 FMT_SMF.1	no yes

Table 1: SFR Functional Requirements Dependencies Analysis

The dependency of FDP_ROL.1 is not satisfied since the rollback functionality defined is not bound to an access control policy. It is rollback performed by the TSF to a valid version of the TSF.

The dependency of FMT_MTD.1 on FMT_SMR.1 is not satisfied since the update may be triggered in a way that does not involve a role of the TOE. For example the update may be triggered by an external signal or an automatic check where the TOE identifies that an update to TSF code and data is available and then performs the update. In this case no TSF-defined role is involved in this process.

Chapter 8 Application Notes

A secure update process for TSF code and/or data ensures that the software and data loaded as part of the update process comes from a trusted source authorized to provide such an update and that the integrity of the data received as part of the update process has not been tampered with.

Such a process requires that the TSF has some means to verify the integrity and authenticity of the data received as part of the update process. Those means often rely on specific data like public keys. I many cases this data itself can be managed by adding new data like public keys or by replacing this data. If such a management process exists it has to be described using the optional SFRs defined in this PP module.

This PP module does not prescribe how the authenticity and integrity verification is performed. It just requires that this process cannot be forged by a threat agent which requires that it is not possible for the threat agent to deliberately create a modified version of a part of the code and data loaded during the update process that passes the authenticity and integrity verification mechanism without such modifications being detected. If required, this also includes a mechanism to detect replay attacks.

In many cases the integrity verification mechanism will rely on some secret like a cryptographic key to satisfy the condition mentioned above. In this case this secret also needs to be kept confidential such that a threat agent is not able to deduce information about the value of that secret.

Chapter 9 List of Abbreviations

Abbreviation	Translation
CC	Common Criteria
PP	Protection Profile
HW	Hardware
SW	Software
OS	Operating System
SFR	Security Functional Requirement
SAR	Security Assurance Requirement

Chapter 10 Bibliography

[1] Common Criteria for Information Technology Security Evaluation. Part 1: Introduction and general model. Version 3.1, Revision 5. April 2017. CCMB-2017-04-001

[2] Common Criteria for Information Technology Security Evaluation. Part 2: Security Functional Components. Version 3.1, Revision 5. April 2017. CCMB-2017-04-002

[3] Common Criteria for Information Technology Security Evaluation. Part 3: Security Assurance Components. Version 3.1, Revision 5. April 2017. CCMB-2017-04-003

[4] Common Criteria for Information Technology Security Evaluation. Evaluation Methodology. Version 3.1, Revision 5. April 2017. CCMB-2017-04-004