MILS Complete Separation Platform Protection Profile

MILS CSP PP

Dr. Igor Furgel
T-Systems International GmbH
Bonn, Germany
igor.furgel@t-systems.com

Viola Saftig
T-Systems International GmbH
Bonn, Germany
viola.saftig@t-systems.com

Abstract — **This paper considers the applicability of Common Criteria (CC) to 'minimally necessary (complete) separation platform' comprising underlying hardware and separation (virtualisation) kernel and shares our experience in modelling the security policy for the MILS separation platform. We present our results on the MILS Protection Profile for Complete Separation Platform (MILS CSP PP).**

Keywords—Common Criteria, Protection Profile, Embedded Systems, Underlying Hardware Platform, Operating System, Separation Kernel, MILS (Multiple Independent Levels of Security), Virtualization, Hypervisor

I. Introduction

Based on embedded devices, cyber-physical networks have become an important part of our society and, hence, impact our society and behaviour of individuals. Even more: there are strong indications of becoming cyber-physical networks a basis for cyber-social networks. Next generations of aircrafts, cars and other vehicles will be tightly interconnected with each other, with the internet, and other infrastructures. The same holds for many areas of our life such as healthcare, energy, finance, and mobile (so called critical infrastructures).

When embedded devices are networked, lack of security can obviously cause mutual problems. In order to provide trustworthiness by security and safety and exclude devastating, unauthorized use of critical systems, it is essential to enforce required security and safety policies in a certified fashion.

A practical issue faced by system integrators and operators of embedded systems is how to build up and operate devices with a mix of critical and of unknown (and untrustworthy) applications in a secure and reliable way. One approach is the Multiple Independent Levels of Security (MILS) security architecture aiming high-assurance and based on the concepts of resource separation and controlled information flow. The heart of the MILS architecture is a small virtualization platform offering a secure decomposition of complex embedded systems into logically independent components.

A challenge particular to the MILS architecture as a layer sitting on embedded hardware and providing a partitioning platform is that its TOE (Target of Evaluation) boundary may comprise very different system constellations defined by executable user data. Thus, in a Protection Profile for MILS separation platform (MILS CSP PP), a generic, but clear description is mandated (1) for the components of a MILS system and (2) for the obligations during system integration, while determining the operational environment and selecting concrete components.

This paper considers the applicability of Common Criteria (CC) ([1], [2], [3], [4]) to 'minimally necessary (complete) separation platform' comprising underlying hardware and separation (virtualisation) kernel and shares our experience in modelling the security policy for the MILS separation platform in the 'Common Criteria Protection Profile: Multiple Independent Levels of Security: Complete Separating Platform' [5] (MILS CSP PP).

The MILS Protection Profile for Complete Separation Platform (MILS CSP PP) is intended to be part of a set of MILS PPs that should comprise also other PPs regarding MILS architecture, like 'Protection Profile: Multiple Independent Levels of Security: Operating System' [10] addressing only Operating System without hardware platform and a PP for the entire integrated system.

The full MILS CSP PP [5] is not published yet but is open for comments and usage. If interested please send an email to the authors.

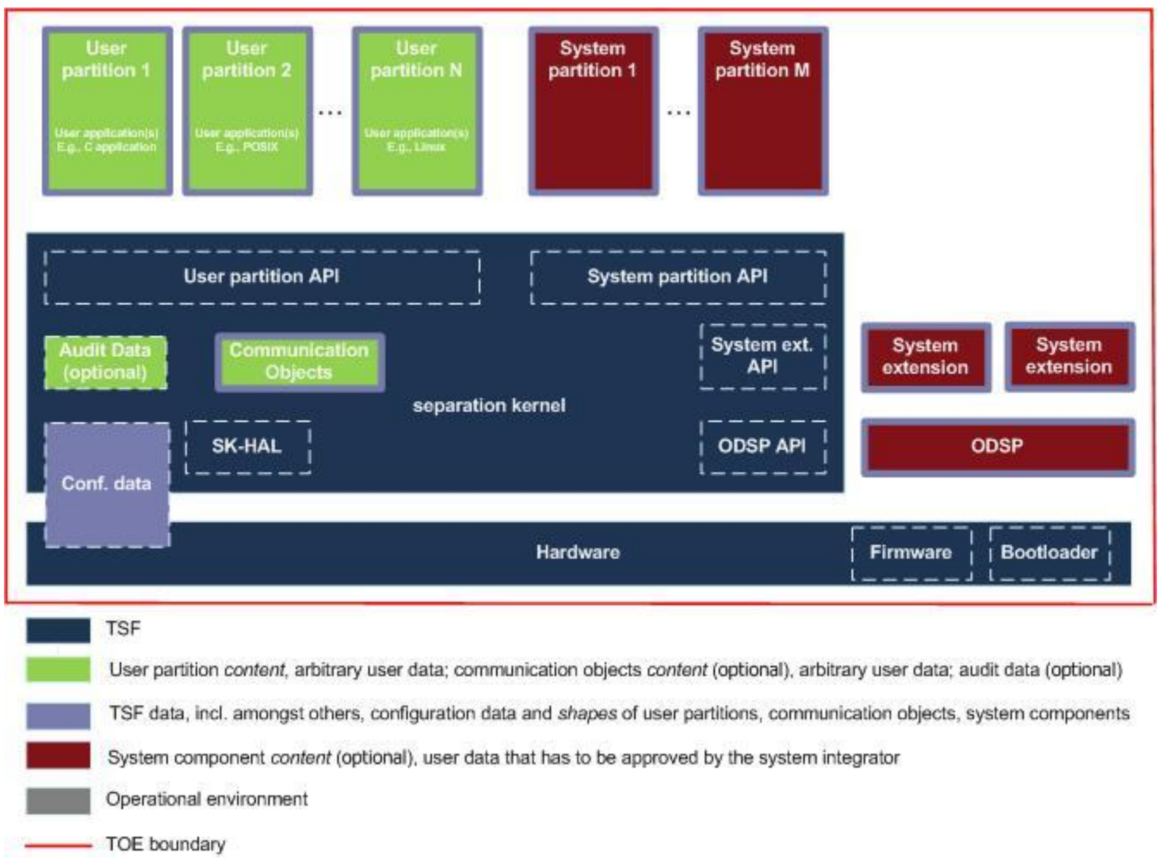
User partition 1
User application(s) E.g., C application
User partition 2
User application(s) E.g., POSIX
User partition N
User application(s) E.g., Linux
System partition 1
System partition M
User partition API
System partition API
Audit Data (optional)
Communication Objects
System ext. API
System extension
System extension
separation kernel
SK-HAL
ODSP API
ODSP
Conf. data
Hardware
Firmware
Bootloader
TSF
User partition content, arbitrary user data; communication objects content (optional), arbitrary user data; audit data (optional)
TSF data, incl. amongst others, configuration data and shapes of user partitions, communication objects, system components
System component content (optional), user data that has to be approved by the system integrator
Operational environment
TOE boundary

Fig. 1. Overview of the TOE [5]

II. MILS CSP PP Overview

The Target of Evaluation (TOE) addressed by the MILS CSP PP [5] is a special kind of operating system and an underlying hardware platform, that together allow to effectively separate different applications running on the same separating platform from each other.

The TOE can host user applications that can also be operating systems. User applications can even be malicious, and even in that case the TOE ensures that malicious user applications are neither harming the TOE nor other applications in other partitions. The TOE may be used e.g. as part of embedded systems.

The TOE is intended to be used as an integrated component (the complete separating platform comprising the underlying hardware and the separation kernel) in MILS systems. MILS (Multiple Independent Levels of Security) systems are explained in [11], [12] and [13].

The TOE controls usage of memory, devices, processors, and communication channels to ensure complete separation of user applications and to prevent unexpected interference between user applications. The TOE enforces restrictions on the communication between the separated user applications as specified by the configuration data.

III. TOE Architecture

The physical scope of the TOE is depicted in Figure 1 and comprises the underlying hardware, the separation kernel, some optional system (i.e. trusted by system integrator) components and some user (non-trusted) components (see CSP PP [5], Figure 1).

The underlying hardware platform and the separation kernel provide together the TOE Security Functionality (TSF). They execute and operate the TOE, by implementing mechanisms to assign resources to partitions, providing the execution environment for applications, and implementing communication between partitions as defined by the configuration data.

The separation kernel provides Application Programming Interfaces (APIs) to user partitions and system partitions as well as APIs to system extensions and on-board device support package (ODSP). An on-board device support package is a special purpose Hardware Abstraction Layer (HAL) and may contain a set of drivers for specific hardware components and represent a system application. A Separation Kernel Hardware Abstraction Layer (SK-HAL) provides specific low-level functionality for each supported CPU architecture. In operational use, the TOE always contains only one SK-HAL.

A user partition contains user applications and/or data being executed and/or stored in a user partition. User applications can be arbitrary and even malicious. User applications use the user partition API of the separation kernel. The content of a user partition does not have to be approved by the system integrator. The content of a user partition can be exchanged without changing the separation kernel binary image, the content of any other partition or the content of a system component of the TOE (see Section 1.3.4.2 of CSP PP [5]).

A system partition contains applications and/or data approved by the system integrator. An application in a system partition is a system application and uses the system partition API of the separation kernel. The content of a system partition can be exchanged without changing the separation kernel binary image, the content of any other partition or the content of a system component of the TOE.

A system extension contains a software component (a system application) approved by the system integrator and coupled with the separation kernel via the system extension API. A system extension can provide specific functionality to applications within partitions only under supervision of the separation kernel. A system extension can be exchanged without changing the separation kernel binary image, the content of any other partition or the content of a system component of the TOE.

IV. Assets

The very high-level security policy defined by this PP can be formulated as follows:
user applications running within a user partition (a green box) shall be separated (i) from applications running in the context of all other partitions (other green boxes and red boxes) and (ii) from the virtualization platform itself (blue boxes) in a controlled way, whereby the TSF (TOE security functionality) is the only controlling entity.

Following the CC formalities, the PP defines a set of primary assets (i.e. values being really important for the risk owner and to be protected by the TOE itself) in order to describe this high-level security policy in the CC language (see CSP PP [5], Table 1). The assets are listed in Table 1 with a short description and the generic security properties which are to be maintained by the TOE as long as the TOE is operational.

For the description of the assets the following two terms are defined: system component and communication object.

A system component is a system partition, system extension or an ODSP and contains user data approved by the system integrator.

A communication object is an object exposed to one or multiple partitions with access rights as defined in the configuration data. Hence, partitions can communicate with each other under the supervision of the TOE's separation kernel only by means of communication objects.

TABLE I. PRIMARY ASSETS

Asset Name	Description	Generic Security Properties
User partition content	User partition content is user applications and/or data being executed and/or stored in a user partition.	confidentiality, integrity
Communication object content	Communication object content is the content of a communication object and exchanged (received/read and sent/written) between partitions.	confidentiality, integrity
System component content	System component content are system applications and/or data being executed and/or stored in a system component (a system partition, a system extension or the on-board device support package).	confidentiality, integrity

So called secondary assets, i.e. TSF and TSF configuration data enforcing the System Security Policy (SSP) as defined by the System Integrator, are described in CSP PP [5], Table 2 and summarized in Table 2 below.

TABLE II. SECONDARY ASSETS

Asset Name	Description	Generic Security Properties
User partition resources	User partition resources comprise physical memory space and allocated CPU time for each CPU. Resources are assigned according to the SSP.	availability
User partition shape	A user partition shape contains a set of security attributes according to the SSP assigned to a user partition that links its user partition resources and its user partition content. A user partition shape contains the following security attributes: a unique partition identity, a flag indicating that the partition is a user partition (i.e. the role for all applications in the partition), and the resource usage data (i.e. here partition resource usage data), SSP enforcement data. User partition shapes can contain also other, security irrelevant data, e.g. information on optimising virtualised guests that is not security relevant.	confidentiality, integrity
Communication object resources	Communication object resources are memory space.	availability

Asset Name	Description	Generic Security Properties
	Resources are assigned according to the SSP	
Communication object shape	A communication object shape contains a set of security attributes according to the SSP assigned to a communication object, which links its communication object resources and its communication object content. A communication object shape contains, amongst other, a unique communication object identity and the resource usage data (i.e. here communication object resource usage data).	confidentiality, integrity
System component resources	Resources of a system component comprise physical memory space and allocated CPU time for each CPU. Resources are assigned according to the SSP.	availability, confidentiality, integrity
System component shape	A system component shape contains a set of security attributes according to the SSP assigned to a system component that links its system component resources and its system component content. A system component shape of a system partition also contains, amongst other a flag indicating that the partition is a system partition, and the resource usage data (i.e. here partition resource usage data).	confidentiality, integrity
Configuration data	Configuration data are data used by the TOE to enforce the SSP.	confidentiality, integrity
System application API	The system application API is an interface to functions of the TSF available for system applications.	availability (in the sense of 'executability') only for system applications

V. Roles

In order to describe the security policy in a proper way, the PP defined the following roles (see CSP PP [5], Table 3):

a) *User application*

A *user application* is any application within a user partition. A user application is allowed to use only the TOE user partition API.

For each instantiation of this subject the TOE assigns a unique subject identity.

b) *System application*

A system application is any application within a system partition, a system extension, or the on-board device support package (ODSP). Only a system application in a system partition is allowed to use the TOE system partition API. Only a system application in a system extension is allowed to use the TOE system extension API. Only a system application in the ODSP is allowed to use the TOE ODSP API.

For each instantiation of this subject the TOE assigns a unique subject identity.

c) *System integrator*

A system integrator is a person trusted to (re-)configure and integrate the TOE. This includes identifying system partitions and user partitions and assigning applications into partitions. System integrator may (and usually do) act on behalf of an organization.

d) *System operator*

A system operator is a person trusted to (re-)install, stop, start, restart, or access (also physically) the TOE in the field. System operator may (and usually do) act on behalf of an organization.

e) *Attacker*

An attacker is a threat agent (a person or a process acting on his/her behalf) trying to undermine the TOE security policy defined by the current PP and, hence, the System Security Policy. The attacker especially tries to change properties of the assets having to be maintained according to the TOE security policy defined by the current PP (see Table 1 and Table 2 above and in Section 3.1.1 of CSP PP [5]). The attacker is assumed to possess an at most high attack potential.

Note that the TOE security policy defined by the current PP only addresses attacks carried out by user applications and does not address any physical attacks. *All attacks from other sources than user applications shall be averted by the TOE operational environment*.

The TOE shall be able to enforce the System Security Policy as defined by the System Integrator. The TOE environment shall support this, especially in the 'system integration' life phase of the TOE in the responsibility of System Integrator (see CSP PP [5], Figure 2).

The security critical TOE environment touches on the following aspects:

- Definition of a System Integration Policy; it includes decisions on all the single components (also hardware) to be integrated into the TOE and on the TOE configuration as appropriate for a concrete business model of the TOE issuer / risk owner,

- Performing system integration according to the Policy, and
- Issuing Operational Policy to the System Operator, if appropriate.

The related obligations of the Systems Integrator are represented in the PP as an Organizational Security Policy.

VI. Security Requirements

The related security requirements on the TOE (i.e. security functional requirements (SFR) to be enforced by TSF and security assurance requirements defining the scope and the rigour of the evaluator's verification work) are defined in a detailed way in chap. 6 of CSP PP [5] and are summarized in the following.

This PP claims conformance to the CC assurance package EAL5 augmented by AVA_VAN.5 (resistance against high attack potential).

The related TOE implementation and guidance in the TOE User Manuals shall ensure that the TOE separation kernel gets *exclusively* executed on the TOE hardware platform. Hence, the assurance components ADV_ARC.1 and AGD_PRE.1 shall, amongst other, cover this aspect.

In order to give an overview of the SFRs in the context of the security services offered by the TOE, security functional groups are defined and allocated to the functional requirements in Table 3. SFRs which are always used together are grouped by '{}' and SFRs whose fulfilling might need a direct support by the TOE hardware are tagged by 'HW'.

TABLE III. Security Requirements

Security Functional Group	Security Functional Requirements (SFRs)
SFG_SSA: Separation in space of applications hosted in different partitions from each other and from the TOE operating system.	{FDP_ACC.2/AS.USER_PART_CONT, FDP_ACF.1/AS.USER_PART_CONT}$_{HW}$, {FDP_ACC.2/AS.SYS_COMP_CONT, FDP_ACF.1/AS.SYS_COMP_CONT}$_{HW}$, {FDP_IFC.2, FDP_IFF.1}, FDP_IFF.5, FRU_RSA.2/AS.USER_PART_RES Supported by: FIA_UID.2, all selected components of the class FMT, all selected components of the class FPT
SFG_STA: Separation in time of applications hosted in different partitions from each other and from the TOE operating system	{FDP_ACC.2/AS.COMMUN_OBJ_CONT, FDP_ACF.1/AS.COMMUN_OBJ_CONT}, {FDP_IFC.2, FDP_IFF.1}, FDP_IFF.5, FDP_RIP.2$_{HW}$, FRU_PRS.1, FRU_RSA.2/AS.USER_PART_RES Supported by: FIA_UID.2, all selected components of the class FMT, all selected components of the class FPT
SFG_COM: Provision and management of communication objects	{FDP_ACC.2/AS.COMMUN_OBJ_CONT, FDP_ACF.1/AS.COMMUN_OBJ_CONT}, {FDP_IFC.2, FDP_IFF.1}, FDP_IFF.5, FRU_RSA.2/AS.COMMUN_OBJ_RES Supported by: FIA_UID.2, all selected components of the class FMT, all selected components of the class FPT
SFG_MAN: Management of and access to the TSF and TSF data	FIA_UID.2, all selected components of the class FMT$_{HW}$
SFG_SPT: TSF self-protection and accuracy of security functionality	FPT_FLS.1, FPT_RCV.2 Supported by: FIA_UID.2, all selected components of the class FMT

Acknowledgment

Part of this PP is based on SKPP [7], [8], OSPP [9], HASK-PP [6], MILS PP: Operating System [10]; the security functional group approach is from [14].

References

[1] Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model; Version 3.1, Revision 4, September 2012, CCMB-2012-09-001

[2] Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Components, Version 3.1, Revision 4, September 2012, CCMB-2012-09-002

[3] Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Requirements; Version 3.1, Revision 4, September 2012, CCMB-2012-09-003

[4] Common Methodology for Information Technology Security Evaluation, Evaluation Methodology; Version 3.1, Revision 4, September 2012, CCMB-2012-09-004

[5] Common Criteria Protection Profile: Multiple Independent Levels of Security: Complete Separating Platform, Igor Furgel, Viola Saftig, Version 1.00, 16.06.2016

[6] Protection Profile for High-Assurance Security Kernel, Bundesamt für Sicherheit in der Informationstechnik (BSI) and Sirrix AG security technologies, Version 1.14, June 2008

[7] U.S. Government Protection Profile for Separation Kernels in Environments Requiring High Robustness, Information Assurance Directorate,. Version 1.03, June 2007

[8] Separation Kernel Protection Profile revisited: Choices and rationale, Timothy E. Levin, Thuy D. Nguyen, Cynthia E. Irvine, Michael McEvilley, 4th Annual Layered Assurance Workshop (LAW), 2010

[9] Operating System Protection Profile, Stephan Mueller, Gerald Krummeck, Helmut Kurth, 2010

[10] Common Criteria Protection Profile: Multiple Independent Levels of Security: Operating System, Igor Furgel, Viola Saftig, EURO-MILS Consortium, Version 2.03, 31.03.2016, www.euromils.eu/publications

[11] Design and verification of secure systems, 8th ACM Symposium on Operating System Principles, John Rushby, 1981

[12] The MILS architecture for a secure global information grid, W. Scott Harrison, Nadine Hanebutte, Paul Oman, Jim Alves-Foss, CrossTalk 18 (10), p. 20–24, 2005

[13] The MILS architecture for high assurance embedded systems, Jim Alves-Foss, W. Scott Harrison, Paul Oman, Carol Taylor, International Journal of Embedded Systems 2 (3/4), p. 239-247, 2006

[14] How to Create a slim and comprehensive PP: The Frame Approach, International Common Criteria Conferrence (ICCC), Igor Furgel, 2013, https://www.fbcinc.com/e/iccc/presentations/T3_D2_12pm_Furgel_How_to_create_a_slim_and_comprehensive_PP.pdf