Toward a Medium-Robustness Separation Kernel Protection Profile

Rance J. DeLong
Santa Clara University
Santa Clara, CA
rdelong@engr.scu.edu

Thuy D. Nguyen
Naval Postgraduate School
Monterey, CA
tdnguyen@nps.edu

Cynthia E. Irvine
Naval Postgraduate School
Monterey, CA
irvine@nps.edu

Timothy E. Levin
Naval Postgraduate School
Monterey, CA
levin@nps.edu

Abstract

A protection profile for high-robustness separation kernels has recently been validated and several implementations are under development. However, medium-robustness separation kernel development efforts have no protection profile, although the US Government has published guidance for authoring such a profile.

As a step toward a protection profile, a set of security requirements for medium-robustness separation kernels is proposed. These requirements result from an informal, yet principled, approach. By bracketing the problem with appropriate reference points and elaborating a method for interpolating the requirements both a measure of uniformity and a basis for further discussion are achieved. Our reference points include the high robustness protection profile, the existing medium robustness consistency instruction, and our familiarity with the nuances of separation kernels.

This practitioner-oriented study is intended to advance the prevailing practices for commercial software development, which presently falls far short of the rigor needed for either high-robustness or medium-robustness systems. These requirements represent an incremental improvement in the pursuit of secure software — and is intended to be a step forward on the road to higher assurance.

1 Introduction

The separation kernel [Rus81] has emerged as a promising foundation for the construction of highly secure systems [VBC+05]. In such applications a separation kernel must exhibit high robustness in the face of attacks by resourceful adversaries against high-value resources under its control.

Robustness

The Common Criteria addresses only functionality and assurance, not robustness. The U.S. Department of Defense defines three level of robustness: high, medium and basic. In this context robustness is "a characterization of the strength of a security function, mechanism, service or solution, and the assurance (or confidence) that it is implemented and functioning correctly." [DOD03] The robustness of a TOE represents the TOE's ability to mitigate security threats in its operational environment. High robustness requires the security mechanisms to "provide the most stringent protection and rigorous security countermeasures" whereas medium robustness imposes requirements for "layering of addtitional safeguards above good commercial practices." [DOD03] Best commerical practices are considered as basic robustness.

The Separation Kernel Protection Profile (SKPP) [SKP07] provides a set of security functional requirements (SFRs) and security assurance requirements (SARs) for separation kernels that will be employed in environments requiring high robustness. It admits implementations ranging from statically-configured partitioning kernels with coarse-grained information flow control enforcement through dynamically-configured kernels with a richer set of exported resources and corresponding fine-grained information flow control policy enforcement [LIN06].

Not every environment, however, requires such a high degree of robustness, since physical access constraints may guarantee a level of trustworthiness of individuals having access to the system. In such applications a medium-robustness separation kernel (MR SK) may suffice. Nevertheless, the prospect of using a common set of components and approaches to security engineering problems provides motivation for the existence of separation kernels that are largely feature-comparable to their high-robustness counterparts, but which are required to exhibit only medium robustness.

The U.S. Government has recognized a need for such a class of separation kernels, as evidenced by at least two developments: the publication by NSA of guidence for the application of both high- and medium-robustness separation kernels [NSA05b], and the determination by some DoD programs of the adequacy of a medium-robustness separation kernel for certain applications.

A proper protection profile (PP) for medium-robustness separation kernels would present both SFRs and SARs derived by a methodical analysis of the security environment and security objectives following the model of the Common Criteria [CC205].

This study proposes a set of requirements for medium-robustness separation kernels. Though informally derived, in contrast with the detailed analysis and justification required in a PP, these requirements are based on an interpolation of reliable sources informed by our familiarity with separation kernel requirements. We hope that providing this study can facilitate and provide consistency among ongoing development efforts, as well as offer a stepping stone to a PP. In addition, the separation kernel is one of many potential targets of evaluation that could exist in both high-robustness and medium-robustness implementations, hence a viable repeatable method for "requirements interpolation" could provide wider benefit.

2 Methodology

We wanted to study the security requirements for separation kernels suitable for deployment in environments requiring medium robustness, without taking on the considerable commitment of developing a protection profile.

We hypothesized that, given knowledge of the validated high-robustness SKPP, of medium-robustness consistency guidance, and of the nuances of separation kernels, then it would be possible to arrive systematically at a good approximation of the requirements for a medium-robustness separation kernel without incurring the expense of PP development.

The method should establish the reference points and the reasoning to be applied to allow interpolation of each requirement for medium robustness. Determining this *a priori* would reduce the variance of discretion applied among requirements. If a result appeared unsatisfactory, it could be analyzed to determine why, and then the method tuned and reapplied.

A strategic choice was to use the rationale provided in the SKPP as a key reference, because it is the most detailed written repository of knowledge concerning what makes a separation kernel unique. By applying rationale similar to that used in the SKPP, and making only the necessary changes while adjusting for the reduced assurance level, it is possible to have reasonable confidence that this informally derived set of requirements is a close approximation to that obtainable by a more rigorous analysis.

The methodology involved the following steps:

1. Collect relevant and documentation sources to *consider* for medium robustness guidance and, based on their applicability to this study, choose the final set to be *relied upon.*

2. Determine whether any security functional requirements in the SKPP could be dispensed with outright, or weakened, in a medium-robustness separation kernel.

3. Decide and finalize the functional requirements for a medium-robustness separation kernel, giving preference to functional interchangeability with the high-robustness separation kernel

4. Consider, in turn, each assurance family identified in the SKPP.

5. Identify an appropriate assurance component for each family based on the decision process detailed below in the Security Assurance Requirements section.

The functional and assurance requirements will be enumerated in later sections. The remainder of this section discusses the selection of sources used for the activity.

The assurance/robustness guidance documents identified for initial consideration were:

1. Separation Kernel Protection Profile (SKPP) [SKP07]

2. Common Criteria (CC) [CC205]

3. US Government Protection Profile for Multilevel Operating Systems in Environments Requiring Medium Robustness (MLOSPP)[MLO07]

4. IA Guidance for Systems Based on a Security Real-Time OS (IAG) [NSA05b]

5. Medium Robustness Consistency Instruction Manual (MR-CIM)[NSA05a]

The IAG recommends the use of "a Medium Robustness SRTOS[1] or a High Robustness SRTOS, depending on the scenario and a variety of factors." Further, it states that a PP should comply with the MR-CIM and, as a starting point, use the security requirements from the SKPP and the assurance requirements from the MLOSPP.

A multilevel operating system is very different from a separation kernel. The MLOSPP describes a full-featured operating system with label-based security policy enforcement. The SKPP describes a minimal operating system that lacks not only label-based security but most of the services required of an OS meeting the MLOSPP. A separation kernel could be used as the foundation for implementing the features of a multilevel operating system, and must have at least the strength of function of any mechanism it is used to support. A version of the MLOSPP had been consulted as the SKPP was being developed, and its influence has already been distilled and filtered through the SKPP refinement process. Given the other more appropriate resources at our disposal we chose not to directly rely further upon the MLOSPP for the present exercise.

The IAG-provided guidance regarding medium-robustness separation kernel requirements is indirect: it merely cites other documents as sources for guidance. The documents cited are among those we considered, and the approach suggested by the IAG is very similar to the one described here, with the exception of the exclusion of the MLOSPP.

3 Security Functional Requirements

The SKPP describes a broad class of separation kernels. It is assumed that a medium-robustness separation kernel would be employed in a fashion architecturally similar to its high-robustness counterpart [NSA05b], though in a more sheltered environment. A medium-robustness separation kernel protection profile should reflect this assumption, as it engenders a commonality of components and approaches across the assurance spectrum, fostering cost savings and adaptability to changing environmental requirements.

For situations in which a SK security architecture is developed for an environment requiring medium robustness and then is later applied to an environment requiring high robustness, it would be advantageous if the medium-robustness separation kernel could be replaced by a high-robustness separation kernel with little or no architectural change. Therefore, it is proposed that a medium-robustness separation kernel have SFRs not substantially different from a high-robustness separation kernel, with the minor exceptions noted in the following section. Thus, the greatest difference between a high-robustness separation kernel and a medium-robustness separation kernel would be the SARs.

[1]The IAG defines an SRTOS as "a separation kernel-based Real-Time Operating System that has undergone an appropriate security evaluation." In this study, such an operating system is generically referred to as a "separation kernel."

4 Security Assurance Requirements

The security assurance requirements for evaluation of a medium-robustness separation kernel should be less demanding than those of the high-robustness SKPP. Table 1 summarizes the proposed SARs using SKPP nomenclature, providing information from the source documents for comparison and a reference to the discussion in the following sections. According to convention, component numbers not in parentheses (e.g., "3") indicate an unmodified component from the Common Criteria catalog of SARs, while those in parentheses (e.g., "(3)") indicate an explicit requirement. Numbering of explicit assurance components can be misleading. "(1)" is not necessarily a less demanding requirement than that represented by a "3," or that represented by an explicit requirement "(2)" in another document. Some authors start numbering explicit requirements within a family starting at 1, while others use the number of the CC component most closely matching the explicit component. The " x' " and " x* " designations represent a decrease in the component leveling defined by the SKPP and MR CIM, respectively. The rationale for these changes is provided in the subsections associated with the corresponding families. The EAL 4 and EAL 6 columns represent the security assurance requirements in the standard package for each EAL given in Version 2.3 of the Common Criteria. The SKPP (HR) column gives the SARs from Version 1.03 of the SKPP. The MR CIM column gives the generic medium robustness requirements recommended by the Consistency Instruction Manual.

Though many of the MR SK requirements may correspond to those of MR CIM, a wholesale adoption of the MR CIM requirements is not appropriate for a separation kernel. Special considerations arise from the nature of a separation kernel TOE *qua* separation kernel, and these considerations apply generally to a MR SK as well as to one of high robustness, though the degree to which they may apply must be determined. These considerations played a role in defining the requirements presented here for a medium robustness separation kernel.

In some cases the medium-robustness requirement is derived in a similar manner to that of the corresponding SKPP requirement, though placed lower in the assurance hierarchy. As an example, consider the Functional Specification (ADV_FSP) family. The CC EAL 6 package specifies component "3" (ADV_FSP.3). The SKPP specifies ADV_FSP_EXP.4, a tailored version of component "4," while our medium-robustness requirement replaces the CC EAL 4 component "2" with a tailored version of component

Table 1. Security Assurance Requirements

Assurance Class	Assurance Family	EAL6 CCv2.3	SKPP (HR)	EAL4 CCv2.3	MR CIM	MR SK	MR SK Comment	See Section
Config Mgmt	ACM_AUT	2	2	1	1	1	MR CIM	§5.2
	ACM_CAP	5	5	4	4	4	MR CIM	
	ACM_SCP	3	3	2	2	2	MR CIM	
Delivery and Operation	ADO_DEL_EXP	2	(2)	2	2	(2)	NIST crypto	§5.3.1
	ADO_IGS	1	1	1	1	1		§5.3.2
Development	ADV_ARC_EXP		(1)		(1)	(1)	MR adjusted	§5.4.1
	ADV_CTD_EXP		(1)			(1)	SKPP	§5.4.2
	ADV_FSP_EXP	3	(4)	2	1	(3)	semiformal	§5.4.3
	ADV_HLD_EXP	4	(4)	2	1	(4)	SKPP	§5.4.4
	ADV_IMP_EXP	3	(3)	1	2	2	MR CIM	§5.4.5
	ADV_INI_EXP		(1)			(1)	SKPP	§5.4.6
	ADV_INT_EXP	2	(3)		(1)	(1)	MR CIM	§5.4.7
	ADV_LLD_EXP	2	(2)	1	(1)	(1)	MR CIM	§5.4.8
	ADV_LTD_EXP		(1)			(1)	SKPP	§5.4.9
	ADV_RCR_EXP	2	3	1	1	2	semiformal	§5.4.10
	ADV_SPM_EXP	3	3	1	1	3	formal	§5.4.11
Guidance Documents	AGD_ADM_EXP	1	(1)	1	1	(1)	SKPP	§5.5
	AGD_USR	1	1	1	1	1		
Life Cycle Support	ALC_DVS	2	2	1	1	1	MR CIM	§5.6
	ALC_FLR		3		2	2	MR CIM	
	ALC_LCD	2	2	1	1	1	MR CIM	
	ALC_TAT	3	3	1	1	2	+ impl stds	
Assur. Maint	AMA_AMP_EXP		(1)			(1)	SKPP	§5.7
Platform Assurance	APT_PDF_EXP		(1)			(1')	mod'd SKPP	§5.8.1
	APT_PSP_EXP		(1)			(1')	mod'd SKPP	§5.8.2
	APT_PCT_EXP		(1)			(1')	mod'd SKPP	§5.8.3
	APT_PST_EXP		(1)			(1')	mod'd SKPP	§5.8.4
	APT_PVA_EXP		(1)			(1')	mod'd SKPP	§5.8.5
Tests	ATE_COV	3	3	2	2	2	MR CIM	§5.9
	ATE_DPT	2	3	1	2	2	MR CIM	
	ATE_FUN	2	2	1	1	1	MR CIM	
	ATE_IND	2	3	2	2	2	MR CIM	
Vulnerability Assessment	AVA_CCA_EXP	2	(2)		2	(1*)	interpartition	§5.10.1
	AVA_MSU	3	3	2	2	2	MR CIM	§5.10.2
	AVA_SOF	1	1	1	1	1	MR CIM	§5.10.3
	AVA_VLA_EXP	4	(4)	2	3	3	MR CIM	§5.10.4

"3." In a very few cases, the specified medium-robustness requirement is identical to that specified in the SKPP. Specific considerations influencing the determination of appropriate components are discussed in the following section.

5 Discussion of the Assurance Requirements

The following subsections describe the rationale used to derive the MR SK assurance requirements for each assurance class. In cases where explicit requirements from the SKPP are applicable to the MR SK, excerpts from the SKPP rationale for those explicit requirements are included.

5.1 A Note on Semiformal Style

It was necessary to define an appropriate guideline for "semiformal" for this study since the range of what can qualify as semiformal is very broad. *Informal* is defined as natural language, *formal* is defined as a restricted syntax language with formal semantics, and *semiformal* is anything in between. This would admit natural language with paragraph headings at one extreme and formal specification languages without a formal semantics at the other extreme.

To avoid ambiguity there needs to be a common language among the designer, the implementer, and the evaluator such that requirements can be interpreted the same by all. At a minimum, for semiformal notation we recommend a language with a defined syntax and a well-documented informal semantics that can support reasonably unambiguous compositional reasoning required for correspondence demonstration of evaluation evidences.

5.2 Configuration Management

The ACM class contains three families: CM Automation (AUT), CM Capabilities (CAP), and CM Scope (SCP). The requirements in this class are straightforward. The SKPP directly adopts the standard EAL 6 components for each family in the ACM class. The MR CIM similarly adopts the EAL 4 component. We follow EAL 4 and the MR CIM by requiring ACM_AUT.1, ACM_CAP.4 and ACM_SCP.2.

5.3 Delivery and Operation

The critical nature of delivery is easily overlooked, but it provides a prime opportunity for subversion [Mye80]. In an environment where a separation kernel is used to isolate

levels of sensitive information, though it is accessed only by trustworthy users, undetected subversion during delivery could compromise critical missions. Therefore adoption of the SKPP required components with the modifications discussed below is recommended.

5.3.1 Delivery (ADO_DEL)

While MR CIM requires only component ADO_DEL.2 (detection of modification and of attempts to masquerade as the developer), which is the same for both EAL 4 and EAL 6, it was determined for the SKPP that an explicit requirement was needed. The use of NIST-approved cryptographic signature algorithms and keyed-hash message authentication functions to support trusted delivery of the TOE was required. Starting with the base CC component ADO_DEL.2, elements were added to require the developer to provide documentation for trusted delivery and to demonstrate the use of NIST-validated cryptographic mechanisms in support of their trusted delivery processes, thus providing additional assurance against undetected tampering.

The following application note is suggested for the medium-robustness requirements: though it may be possible to meet the delivery requirements without the use of cryptography, if technical measures are used to satisfy ADO_DEL_EXP.1 and those measures include cryptographic mechanisms, then such mechanisms should implement NIST-approved algorithms, though certification is not required. The additional evaluator action to determine the sufficiency of the strength of mechanism for the trusted delivery mechanism required by the SKPP in ADO_DEL_EXP.2.2E has been dropped.

5.3.2 Installation, Generation and Start-Up (ADO_IGS)

The Common Criteria only defines two components for this family, ADO_IGS.1 and ADO_IGS.2; however, all of the CC standard EAL packages, as well as the SKPP and the MR CIM adopt ADO_IGS.1. Likewise, for medium robustness, only ADO_IGS.1 has been included.

5.4 Development

The ADV class contains the assurance families: Architectural Design (ARC), Configuration Tool Design (CTD), Functional Specification (FSP), High-Level Design (HLD), Implementation Representation (IMP), Trusted Initialization (INI), TSF Internals (INT), Low-Level Design (LLD), Load Tool Design (LTD), Representation Correspondence (RCR), and Security Policy Modeling (SPM) as described in the following subsections.

5.4.1 Architectural Design (ADV_ARC)

This assurance family is not present in the Common Criteria Version 2.3 but it has been explicitly included in the MR CIM, in other medium-robustness protection profiles, and in the SKPP. In the case of the SKPP, it was recognized that a new assurance criterion was necessary to require assurance evidence specific to the TSF architecture and its ability to protect itself, to support the principle of least privilege for the purpose of damage limitation, and to prevent TSF-internal denial of service by executing in a predictable manner. ADV_ARC_EXP.1 was created to address these requirements. It is worth noting that several assurance elements in ADV_ARC_EXP.1 are related to SFRs. Thus the testable desired behavior of the TOE in terms of functional requirements is precisely defined as are the assurances required to determine that the desired behavior is achieved by the implementation.

For medium robustness, a similar explicit ADV_ARC component should be defined and the assurance levied must be commensurate with the degree of scope, depth and rigor provided by the functional specification, high-level design and the TSF internals description. As stated in the application note for ADV_ARC_EXP.1 in the SKPP, "the architecture design required by this component is at the level of the functional specification and high-level design documentation. The TSF internals description required by ADV_INT_EXP.3 component is at the level of TSF module documentation." This reasoning leads to medium robustness for the adjusted rigor of the functional specification, high-level design, and internals description. These adjustments are subsumed by the wording of ADV_ARC_EXP.1.2C, which states that the architectural design "shall be at a level of detail commensurate with the description of the SFR-enforcing abstractions described in the TOE high-level design documentation."

5.4.2 Configuration Tool Design (ADV_CTD)

This assurance family is not defined in the Common Criteria but explicitly added by the SKPP to address concerns about the validity of the configuration vector(s) that the separation kernel relies on to establish the initial secure state and to enforce the partition flow policy. The configuration vector has a direct bearing on the ability to produce an inductive proof of the basic security theorem[2].

The configuration vector is generated and validated by a configuration tool. Because the configuration tool is part of the TOE but not part of the TSF, it is not subject to most of the ADV documentation SARs. The absence of a CC as-

[2]The Basic Security Theorem establishes that the system never enters an unsecure state. A proof of the BST typically takes the form of an induction on states, comprising a secure initial state(s) and security invariant-preserving transitions on states.

surance family that addresses the assurances for generating the configuration vector and for establishing the correctness of the configuration vector drives the need for an explicit requirement.

This explicit requirement compels the developer to provide a design that can be evaluated to assure that the configuration tool is trustworthy to perform its functions. The wording of the explicit requirement specifies the level of rigor as "informal style at a level of abstraction and detail required in the TOE high-level design document."

5.4.3 Functional Specification (ADV_FSP)

The MR CIM has an explicit component ADV_FSP_(EXP).1 for FSP. Inspection of the SKPP ADV_FSP_EXP.4 reveals that its explicit differences have been influenced by the MR CIM ADV_FSP_(EXP).1. The additional requirements, as described in the SKPP rationale, are necessary to provide the evaluator sufficient information to assess the intended use and behavior of each kernel external interface. Before adding these explicit differences, the SKPP escalated the component from ADV_FSP.3 to ADV_FSP.4, which differs solely by a change from semiformal to formal style. This permits the correspondence demonstration between the security policy model and the functional specification to be formal for a high-robustness separation kernel.

For the medium-robustness separation kernel we employ a similar strategy. We escalate ADV_FSP.2 from CC EAL 4 to ADV_FSP.3, which changes the functional specification from informal to semiformal style, and then add the additional requirements levied by the SKPP. The change to semiformal style works hand-in-hand with the formal ADV_SPM requirement to permit a meaningful semiformal correspondence demonstration.

5.4.4 High-Level Design (ADV_HLD)

The SKPP and the MR CIM both specify explicit requirements for HLD. That of the SKPP is based on CC ADV_HLD.4 but is tailored to take into account the structure of the separation kernel and has fewer elements than ADV_HLD.4. Following this pattern, the medium-robustness requirement would be ADV_HLD_EXP.2, a similarly tailored version of ADV_HLD.2.

The SKPP ADV_HLD_EXP.4 calls for a semiformal style supported by informal explanatory text for the TSF, and an informal style for the non-TSF. It is the difference in rigor between TSF and non-TSF that differentiates the requirements for the high-robustness SK from that of a "best practice" high-level design with typical good software engineering discipline and documentation. We believe that reducing the rigor requirement to informal is acceptable for medium robustness. Note that it may be more difficult to distinguish and separate the TSF from the non-TSF if the design was created without having such differentiation as an objective.

5.4.5 Implementation Representation (ADV_IMP)

For medium robustness we follow the MR CIM by employing ADV_IMP.2.

5.4.6 Trusted Initialization (ADV_INI)

Trusted initialization continues the chain of assurance maintained through distribution and configuration by other requirements, but there was not an assurance family for it in the Common Criteria. Trusted Initialization was explicitly added by the SKPP. Because the TOE must be able to initialize and establish a secure initial state autonomously, without any intervention by authorized administrators, assurances are required for trusted initialization of the TOE when that initialization is accomplished without the aid of authorized administrators. The initialization function is responsible for trusted initialization of the TOE which includes establishing the execution environment for the TSF and establishing the TSF in its initial secure state. Since the initialization function is part of the TOE but not part of the TSF, it is not subject to most of the ADV documentation SARs.

For a medium-robustness separation kernel we adopt the same requirement used in the SKPP. The ADV_INI_EXP.1 requirement is levied in recognition that establishment of a secure initial state is fundamental to a proof of the basic security theorem. Although the initialization function is not part of the TSF, the conversion of configuration vectors into the TSF data must be shown to preserve the semantics of the configuration data.. This requirement provides a design for the initialization function to the extent it is not included in the design of the TSF.

5.4.7 TSF Internals (ADV_INT)

This requirement addresses the internal structure of the TSF. The TSF Internals requirement is a CC component that is one of the primary factors contributing to the conventional wisdom that pre-existing products can only be evaluated to EAL 4 (without being redesigned and reimplemented). The standard CC EAL 4 package does not include an ADV_INT requirement. For medium robustness we follow the MR CIM by employing ADV_INT_(EXP).1. This requirement is substantially more specific than ADV_INT.1 in the CC (the standard component for EAL 5). The software engineering discipline entailed by this explicit requirement may challenge pre-existing implementations not developed with this requirement in mind.

5.4.8 Low-Level Design (ADV_LLD)

For medium robustness we adopt ADV_LLD_(EXP).1 of the MR CIM. This requirement is more specific than ADV_LLD.1 and is apparently intended to work together with ADV_INT_(EXP).1 to enforce more stringent software engineering practices. The presentation style required is still informal.

5.4.9 Load Tool Design (ADV_LTD)

Like the configuration tool, this assurance family is not present in the Common Criteria but is explicitly added by the SKPP.

The ADV_LTD_EXP requirement is levied in recognition that the load tool is a crucial part of the TOE's evaluation because it is part of the chain that establishes the initial state, thus having a direct bearing on the ability to provide an inductive proof of the basic security theorem. Through the ADV class, essential assurance measures are applied to security critical components within the TOE; however, because it is not part of the executable kernel, these measures would not be applied to the load tool. An explicit requirement levied specifically on the load tool requires the developer to provide a load tool design that can be evaluated for assurance that it is trustworthy to perform its functions.

5.4.10 Representation Correspondence (ADV_RCR)

The SKPP escalates the EAL 6 component ADV_RCR.2 to ADV_RCR.3 because the SKPP requires formal FSP and SPM. The FSP and SPM proposed for medium-robustness separation kernels by this work are semiformal and formal respectively. We therefore escalate the medium-robustness requirement to ADV_RCR.2 over the EAL 4 component ADV_RCR.1 (also specified in the MR CIM).

5.4.11 Security Policy Modeling (ADV_SPM)

The CC has an idiosyncracy in the usage of the ADV_SPM class in the standard EAL packages, in that the semiformal component is not used in any EAL. SPM is informal at EAL 4 and formal at EAL 5 and above.

We have chosen to utilize the CC ADV_SPM.3 component, a *formal* security policy model, for the medium-robustness separation kernel. In this way a meaningful semiformal correspondence demonstration can be done between the formal security policy model and the semiformal functional specification.

5.5 Guidance Documents

The Common Criteria defines only one component for each of the families AGD_ADM and AGC_USR. The SKPP creates an explicit component, AGD_ADM_EXP.1 because separation kernel specific considerations result in a number of explicit requirements that must be mirrored in the administrator guidance. For medium robustness we adopt the same requirements as the SKPP, viz., AGD_ADM_EXP.1 and AGD_USR.1.

5.6 Life Cycle Support

The ALC class contains the families: Development Security (DVS), Flaw Remediation (FLR), Life Cycle Definition (LCD), and Techniques and Tools (TAT). The Common Criteria does not utilize the FLR family in any of the EAL packages. The SKPP and the MR CIM both, however, include a component from the FLR family.

For medium robustness we follow the MR CIM for ALC families DVS, FLR and LCD by requiring ALC_DVS.1, ALC_FLR.2, and ALC_LCD.1. For the Techniques and Tools family, however, we believe that the ALC_TAT.1 component required by the MR CIM is too weak for a newly developed TOE, which a separation kernel is likely to be, and instead require ALC_TAT.2.

The specific difference between ALC_TAT.1 and ALC_TAT.2 is that the subset of the implementation defined by the ADV_IMP family (we have specified ADV_IMP.2, following the MR CIM) must comply with explicitly stated implementation standards, and that the evaluator must confirm that the standards have been applied. The Common Criteria neglects to state where those implementation standards are to be defined, so we would add an application note to suggest that the developer provide a Techniques and Tools document that includes the definition of the implementation standards to be applied, as well as the other content items required by ALC_TAT.2.

5.7 Maintenance of Assurance

For the SKPP, the explicit component AMA_AMP_EXP.1 was written to define the requirements for an assurance maintenance plan.

While it may be debatable whether the use of AMA_AMP_EXP.1 is required *only* for the high robustness requirement and not for medium robustness, the benefits of assurance maintenence to the developer of a TOE at any robustness level should be recognized. The reality of product change and the cost of evaluation should make apparent the benefits of minimizing reevaluation cost. A well-written Assurance Maintenance Plan (AMP) effectively permits the evaluators to evaluate "at once" the TOE in all variations that the AMP can successfully justify to not require reevaluation. As a practical matter, this should also be a requirement for medium robustness separation kernels.

5.8 Platform Assurance

The MR CIM specifies that domain separation requirements (FPT_SEP) must be included in a medium robustness TOE, and thus the underlying hardware mechanisms that the TOE depends on to support its security architecture must also be included as part of the TOE. To date, the CC does not include requirements for assessing the assurance of hardware components used to implement a TOE's security functions. Since it is difficult for TOE vendors to produce assurance evidence for hardware at the same level of detail that is required for software, it was necessary to introduce a separate assurance class in the SKPP to provide a framework for establishing the security relevance of commercially-available hardware based on its interaction with software through its interfaces. As noted in the rationale for the explicit Platform Assurance (APT) class, the overall approach is to define platform components in terms of specification instead of identification. This is to address the long-standing problem that specific hardware components identified in a TOE's evaluated configuration become obsolete during or immediately after the TOE's evaluation.

5.8.1 Platform Definition (APT_PDF)

This family requires a description of the platform in terms of platform components that can be obtained and assembled by the end users. The Platform Definition Document (PDD) must include the assembly rules and information about the types, interfaces and security properties of the components to support component-specific security analysis against the SFRs. The CC component leveling for this family is determined based on the details provided in the PDD. For high robustness, the SKPP mandates the highest level, i.e., the PDD must include precise component interface specifications for all platform components in addition to the platform component security analysis. The evaluator is also required, at the highest level, to verify a subset of the interface specifications to ensure that they provide adequate information on component compatibility. Based on the MR CIM guidance that "the level of detail of design documentation and the implementation representation is dependent upon a component's role in security policy enforcement," we propose to relax both of these content and evaluator requirements for medium robustness separation kernels. Specifically, the required details of the component interface specifications can be less rigorous, i.e., interface specifications are only required for components that directly affect the implementation of the policy enforcement SFRs, and the evaluator will not be required to verify the interface specifications for compatibility information.

5.8.2 Platform Specification (APT_PSP)

This family levies requirements on the vendor-supplied specifications of the interfaces provided by the platform components. These specifications are necessary to support functional analysis and vulnerability assessment of the TSF. Three CC component levels are defined for this family. The highest level, as specified in the SKPP, requires complete specifications of all platform interfaces (i.e., external, internal and unused internal interfaces). The middle level only requires a complete specification of the external platform interfaces while the basic level simply requires the identification of the external interfaces. For medium robustness, it is sufficient to downgrade to the middle level because a complete specification of the external interfaces can facilitate a critical examination of the hardware functionality that is externally visible to the TOE software which, in turn, can help support the analysis for design correctness and exploitable vulnerabilities of the TSF.

5.8.3 Platform Conformance Testing (APT_PCT)

Regarding hardware testing, the SKPP makes a distinction between testing at the platform component interface and testing at the TSFI interface. This family addresses the latter whose goal is to ensure that the platform components identified in the PDD function as expected by the TSF software. Testing at the component interface level is covered in the APT_PST family described below. The CC component leveling of this family is based on the scope and rigor of the required tests. To satisfy the basic level, the TOE developer is only required to demonstrate that the components provide the functional features required by a valid hardware platform. The middle and highest levels, on the other hand, focus more on exercising the security features provided by the components. The middle level requires testing of only the security features upon which the TSF depends. The highest level requires testing of all security features that are relied upon by the TSF as well as other hardware components. Since exhaustive test coverage is not required for medium robustness, it is logical to use the middle level for medium robustness separation kernels.

5.8.4 Platform Security Testing (APT_PST)

This family defines requirements for security testing of hardware components to be performed at the component interface level. As noted in its rationale, "the intent of this class is to make deterministic tests of the platform mechanism rather than relying on test coverage arguments at the TSFI level." The shift in the testing focus places a stronger emphasis on security assessment to determine how well the components satisfy the SFRs. Two CC component levels are defined for this class. Only external platform interfaces

are required for the basic level while all external and internal interfaces are required for the second level. Naturally, the SKPP utilizes the second level. Since it is anticipated that a thorough testing of the platform mechanisms that are externally visible can provide enough evidence that the external platform interfaces function correctly and can moderately resist attacks, it seems appropriate to levy the basic level requirements on a medium robustness separation kernel

5.8.5 Platform Vulnerability Assessment (APT_PVA)

This family complements the AVA_VLA family by requiring that hardware vulnerability assessment be considered as part of the software vulnerability analysis. Similar to APT_PST, the difference between the two CC component levels defined is based on the type of platform interfaces involved in the assessment, i.e., only external interface (basic level) versus both external and internal interfaces (second level). The same argument used in the APT_PST also applies here and thus, the basic level prevails.

5.9 Tests

The ATE class contains the assurance families: Analysis of Coverage (COV), Depth of Testing (DPT), Functional Testing (FUN), and Independent Testing (IND). In each of these families the medium-robustness separation kernel requirements adopt those of the MR CIM.

5.10 Vulnerability Assessment

The AVA class contains the assurance families: Covert Channel Analysis (CCA), Misuse (FLR), Strength of Function (SOF), and Vulnerability Analysis (VLA).

5.10.1 Covert Channel Analysis (AVA_CCA)

For the CCA family the SKPP specifies an explicit component AVA_CCA_EXP.2, based on AVA_ CCA.2, but limited to cover only *inter-partition* covert channels. One would expect that the covert channel analysis requirement for MR would cover only inter-partition covert channels, whatever the level of rigor of the search.

The MR CIM, on the other hand, has a different explicit component AVA_CCA_(EXP).2, that requires systematic covert channel analysis of the cryptographic module only. It contains an application note that explains that the TSF interfaces are not covered because it is "considered beyond the scope of effort and cost considered reasonable for COTS medium-robustness products." It goes on to acknowledge that this does increase risk.

Experience has shown that it is likely that the exercise of conducting a covert channel search, even one that is not systematic, may expose channels that can and should be mitigated, and can yield valuable information about the TOE to be captured in guidance documentation.

In applying the MR CIM rationale to the medium-robustness separation kernel, it is clear that that the MR CIM requirement for systematic covert channel analysis of the cryptographic module does not apply because the separation kernel provides no cryptographic services. But there are a few issues particularly relevant to separation kernels.

The purpose of a separation kernel is to control information flow; any other function is arguably incidental. Perhaps special consideration for separation kernels is needed. Even at medium-robustness, some covert channel analysis would be beneficial, such as the informal search specified by AVA_CCA.1, which normally comes into play in the Common Criteria EAL 5 package.

For the medium-robustness separation kernel, the covert channel search could be limited to inter-partition information flow policy, thus creating an explicit component AVA_CCA_EXP.1 analogous to the SKPP's AVA_CCA_EXP.2.

5.10.2 Misuse (AVA_MSU)

For MSU the SKPP follows the EAL 6 component and the MR CIM follows the EAL 4 component. For medium-robustness separation kernels, the component specified by the MR CIM, AVA_MSU.2 Validation of Analysis is adopted.

5.10.3 Strength of Function (AVA_SOF)

The Common Criteria defines only one component for the SOF family. The SKPP and the MR CIM both adopt AVA_SOF.1, as does this study. As noted in the SKPP, these AVA_SOF requirements are only applicable to the additional security requirements defined in the Security Target for which a claim of strength of function is appropriate.

5.10.4 Vulnerability Analysis (AVA_VLA)

For the VLA family the SKPP creates an explicit requirement that modifies AVA_VLA.4 only in the evaluator actions: it requires that an *NSA evaluator* conduct *independent* vulnerability analysis and penetration testing *not building on* developer vulnerability analysis.

The MR CIM elevates the VLA requirement AVA_VLA.2, specified by the EAL 4 package, to AVA_VLA.3 Moderately Resistant, which requires that the vulnerability search be demonstrably *systematic*. This is what we adopt for the MR SK.

6 "Catch-22" Challenges

The absence of a validated PP for medium-robustness separation kernels poses a challenge for developers. The Common Criteria Evaluation and Validation Scheme (CCEVS) Robustness FAQ responds to the question, "can a TOE/ST claim a robustness level without conforming to a PP?" with the answer, "For Medium or High Robustness, this would be theoretically possible if there were an ST Review Board (analogous to the PP Review Board) that would review the ST to ensure that it adheres to the rules set forth in the Consistency Instruction Manuals. There currently is no such group, so there is no way to claim a Medium or High Robustness level without claiming conformance to a PP." Likewise, the IAG position also refers to a security target, which it says should conform to a currently non-existent protection profile. Further, the position that a protection profile acceptable to NSA must be a "U.S. Government Protection Profile," and that such a PP can be developed only by NSA, leads to an effective impasse for developers. The upshot of this dilemma is that either a proper protection profile needs to be developed by NSA, or a non-NSA produced PP would need to be endorsed by NSA.

7 Summary and Conclusions

We have presented security functional and assurance requirements for a medium-robustness separation kernel. Rather than performing a deep protection profile-style analysis of security environment and objectives, we drew upon the SKPP and guidance documents to interpolate the requirements informally, following a methodology that may serve as an example for future efforts to interpolate medium robustness requirements corresponding to future high robustness PPs.

The SKPP differs in important ways from the standard EAL 6 package. This is due to special factors that arise for a separation kernel *qua* separation kernel. These factors must also be given consideration when determining the requirements for a MR SK. Consequently, the MR SK assurance components do not follow the MR CIM in every detail, but in some cases follow the pattern of the SKPP instead.

Although our results are not intended to replace a medium-robustness separation kernel protection profile, they do provide a step in that direction. They adopt virtually *all* of the SKPP functional requirements and have *met or exceeded all* of the MR-CIM (and EAL 4) assurance requirements, on a few items *equalling* the SKPP requirements. Absent a protection profile, the requirements presented herein constitute a conservative basis for proceeding with medium-robustness separation kernel development.

References

[CC205] *Common Criteria for Information Technology Security Evaluation*, August 2005. Version 2.3, CCMB-2005-08-001, 002, 003.

[DOD03] *Department of Defense Instruction, Number 8500.2*, February 6 2003.

[LIN06] Timothy E. Levin, Cynthia E. Irvine, and Thuy D. Nguyen. Least privilege in separation kernels. In *Proceedings of the IEEE International Conference on Security and Cryptography*, Setubal, PT, August 2006.

[MLO07] Information Assurance Directorate, National Security Agency, Fort George G. Meade, MD 20755-6000. *U.S. Government Protection Profile for Multilevel Operating Systems in Medium Robustness Environments, Version 1.91*, March 2007.

[Mye80] P. Myers. *Subversion: The Neglected Aspect of Computer Security*. M.S. thesis, Naval Postgraduate School, Monterey, CA, 1980.

[NSA05a] National Security Agency, Information Assurance Directorate. *Consistency Instruction Manual for development of U.S. Government Protection Profiles for use in Medium Robustness Environments*, February 2005. Release 3.0.

[NSA05b] National Security Agency, Systems Security Engineering, Information Assurance Directorate, Ft. Meade, MD. *Information Assurance Guidance for Systems Based on a Security Real-Time Operating System*, December 2005. NSA SSE-100-1.

[Rus81] J. Rushby. The design and verification of secure systems. In *Eighth ACM Symposium on Operating System Principles*, pages 12–21, Asilomar, CA, December 1981. (ACM *Operating Systems Review*, Vol. 15, No. 5).

[SKP07] Information Assurance Directorate, National Security Agency, Fort George G. Meade, MD 20755-6000. *U.S. Government Protection Profile for Separation Kernels in Environments Requiring High Robustness*, June 2007. Version 1.03.

[VBC+05] W. M. Vanfleet, R. W. Beckwith, B. Calloni, J. A. Luke, C. Taylor, and G. Uchenick. MILS: architecture for high assurance embedded computing. *CrossTalk*, 18:12–16, August 2005.