NCSC-TG-021
Library
No. S235,625

FOREWORD

The National Computer Security Center is issuing the Trusted Database Management System Interpretation as part of the Technical Guidelines Program, through which we produce the "Rainbow Series." In the Rainbow Series, we discuss in detail the features of the Trusted Computer System Evaluation Criteria (DoD 5200.28-STD) and provide guidance for meeting each requirement. The National Computer Security Center, through its Trusted Product Evaluation Program, analyzes the security features of commercially produced and supported computer systems. Together, these programs ensure that organizations are capable of protecting their important data with trusted computer systems.

The Trusted Database Management System Interpretation extends the evaluation classes of the Trusted Computer System Evaluation Criteria to trusted applications in general, and database management systems in particular. It serves as an adjunct to the Trusted Computer System Evaluation Criteria by providing a technical context for the consideration of entire systems constructed of parts and by presenting database-specific interpretation of topics that require direct comment. Thus, it is relevant to applications which support sharing of computer services and resources, and which enforce access control policies. More specifically, it provides insight into the the design, implementation, evaluation, and accreditation of database management systems.

This document will be used for at least one year after the date of signature. During this period the NCSC will gain experience using the Trusted Database Management Systems Interpretation in several evaluations and continue to receive comments on issues of technical accuracy, clarity of exposition, and utility. After this trial period, necessary changes will be made and a revised version issued.

____

PATRICK R. GALLAGHER, JR. April 1991
Director National Computer Security Center

ACKNOWLEDGMENTS

This document represents the combined effort of participants from the research community, industry, and government working over several years. Three major drafts and numerous intermediary versions were produced, reviewed, and commented upon. To name all the contributors would be impossible. To single out a few would be to slight a host of others who gave unstintingly of their time and talent. To all those who contributed to the development and refinement of the fundamental concepts, contributed to the development of the several predecessor versions, and who contributed valuable personal time and invaluable experience in reviewing and commenting on the previous versions, thanks is extended.

TABLE OF CONTENTS

INTRODUCTION

HISTORICAL PERSPECTIVE

The Department of Defense Trusted Computer System Evaluation Criteria (TCSEC), published in 1983 as CSC-STD-001-83, consolidates knowledge about the degree of trust one can place in a computer system to protect sensitive information and organizes this knowledge into usable criteria for system evaluation. The TCSEC was republished as a DoD standard, DoD-5200.28-STD, in December 1985 to provide a means of evaluating specific security features and assurances available in "trusted, commercially available automatic data processing system

The TCSEC's rating scale extends from a minimal to a high level of trust with advanced security features and sophisticated assurance measures. Specific criteria determine each rating level and guide system builders and evaluators in determining the level of trust provided by specific systems. When the rating levels are combined with environmental guidelines and minimum security protection requirements, accreditation decisions for specific installations can be made.

The philosophy of protection embodied in the TCSEC requires that the access of subjects (i.e., processes in a domain) to objects (i.e., containers of information) be mediated in accordance with an explicit and well-defined security policy. At the higher criteria classes, the "reference monitor concept" [1] is an essential part of the system and the security policy is modeled. There are several security policy models that represent the desired behavior of a reference monitor. The Bell-La Padula model [4-6] and its Multics interpretation [3] are commonly used, but not mandated.

The computer security research and development that underpin the TCSEC began in the late 1960s and concentrated on secure operating systems. By the early 1970s initial worked examples had provided a substantial amount of information about building trust into operating systems. Research continued throughout the 1970s to refine mechanisms, features, and assurances needed to provide trusted operating systems.

Multilevel database management security received far less research and development attention than did secure operating systems. This was primarily due to the perception that one could not credibly implement a multilevel secure database management system (DBMS) on top of an untrusted operating system base. However, some research in multilevel secure DBMSs (mostly theoretical) was conducted during the 1970s [15-16], and research has continued to the present [9-14, 17-19, 22, 25-28]. By the mid 1980s, commercially developed, trusted operating systems were becoming available that could provide the basis for hosting secure applications such as multilevel secure DBMSs.

In June 1986, the National Computer Security Center (NCSC) initiated its efforts to address the evaluation of trusted database management systems with an Invitational Workshop in Baltimore, Maryland. Representatives from the research, database vendor, commercial, and government communities met to address issues of database management security. The attendees met to discuss aspects of trust (defined by the TCSEC) that were sufficiently well defined and capable of producing repeatable and objective assessments. The NCSC invited issue papers and prepared a discussion agenda. The issue papers and the subcommittee summaries were published as the Proceedings of the National Computer Security Center Invitational Workshop on Database Security [20].

As an outgrowth of this workshop, the NCSC undertook the task of preparing this Trusted Database Management System Interpretation (TDI) of the TCSEC to focus on the special problems posed by DBMSs. A working group was assembled to draft this Interpretation. Three drafts were produced, with extensive community review and public discussion. This Interpretation is the result of the interaction within the general computer security and database management communities.

SCOPE

The interpretations in this document are intended to be used in conjunction with the TCSEC itself; they apply to application-oriented software systems in general, and database management systems (DBMSs) in particular. Although the interpretations, as noted, are general enough to apply to any software system which supports sharing and needs to enforce access control (e.g., transaction processing systems, electronic mail systems), in the interest of simplicity, the discussion, and thus the terminology, will be directed toward DBMSs.

The interpretations are intended to be

applied primarily to commercially developed trusted DBMSs, but can also be applied to the evaluation of existing non-commercial DBMSs and to the specification of security requirements for DBMS acquisitions.

PURPOSE

This Interpretation of the TCSEC has been prepared for the following purposes:

To provide a standard to manufacturers for security features to build into their new and planned commercial products in order to provide widely available systems that satisfy trust requirements (with particular emphasis on preventing the disclosure of data) for sensitive applications,

To provide a metric with which to evaluate the degree of trust that can be placed in computer systems for the secure processing of classified and other sensitive information, and

To provide a basis for specifying security requirements in acquisition specifications.

With respect to the second purpose for development of the criteria, i.e., providing a security evaluation metric, evaluations can be delineated into two types: (1) evaluations performed on a computer product from a perspective that excludes the application environment; or, (2) evaluations to assess whether appropriate security measures have been taken to permit the system to be used operationally in a specific environment. The former type of evaluation is done by the National Computer Security Center (NCSC) through the Trusted Product Evaluation Program and is called "formal product evaluation."

The latter type of evaluation, that is, one done for the purpose of assessing a system's security attributes with respect to a specific operational mission, is known as a "certification evaluation." A formal product evaluation does not constitute certification or accreditation for the system to be used in any specific application environment. The system security certification and the formal approval/accreditation procedure, done in accordance with the applicable policies of the issuing agencies, must still be followed before a system can be approved for use in processing or handling sensitive or classified information. Designated Approving Authorities (DAAs) remain ultimately responsible for specifying the security of systems they accredit.

The TCSEC and this Interpretation will be used directly and indirectly in the certification process. Along with applicable policy, they will be used directly as technical guidance for evaluation of the total system and for specifying system security and certification requirements for new acquisitions. Where a system being evaluated for certification employs a product that has undergone a formal product evaluation, reports from that process will be used as input to the certification evaluation. Moreover, the National Security Agency plans to publish additional guidelines to assist certifiers and help ensure consistency in certifications of systems processing national security informantion.

STRUCTURE OF THE DOCUMENT

The remainder of the TDI is divided into two parts, plus two appendices and a glossary.

PART 1, "TECHNICAL CONTEXT," presents general information about the evaluation of trusted systems that are constructed of several parts. This discussion is critical to trusted DBMSs built upon trusted operating systems, but is not limited to DBMSs only. It is included in the TDI because it is not currently available in any previously published document. This section reviews the central reference monitor concept, explains the need to evaluate a system built of well-defined parts, and develops the concept of TCB subsets. TCB subsets provide a way of splitting a total TCB along access control policy lines such that an evaluation by parts can be undertaken. PART 1 concludes with an interpretation of those TCSEC requirements which are relevant to an evaluation by parts, and a description of the process of evaluation by parts.

PART 2, "INTERPRETED REQUIREMENTS," provides interpretions of those TCSEC requirements that are either specific to DBMSs (or applications in general), or are particularly relevant to DBMSs. The number of requirements that are treated explicitly is relatively small, because many of the TCSEC requirements apply as stated to database management systems. The requirements treated explicitly are labels, audit, system architecture, design specification and verification, and design documentation.

Appendix A summarizes the interpreted requirements for each TCSEC class and is intended to provide an easy reference for those requiring a listing of requirements for a specific class (e.g., B2).

Appendix B provides discussion of several topics not directly tied to the requirements levied on trusted DBMSs by the interpretation of the TCSEC requirements.

The TDI proper will be supplemented by a Companion Document Series (CDS). The CDS will address a wide spectrum of issues related to trusted DBMSs but which are beyond the scope of this document. Community debate about on-going topics of interest will probably result in gradual extensions of what is known about trusted DBMSs. Thus, volumes in the CDS will be issued both regularly (to document the state of the community debate) and by exception (to record new problems and new solutions).

PART 1

TECHNICAL CONTEXT

TC-1 INTRODUCTION

Modern computing systems are rarely conceived and built by a single organization. Rather, the rule is that systems are constructed by assembling parts hardware, firmware, and software produced independently by various organizations or vendors. This fact introduces a fundamental difficulty into the task of evaluating a "system" for conformance to the trust requirements of the Trusted Computer System Evaluation Criteria (TCSEC). [8] This difficulty stems from the fact that assessment (either evaluation of a product or certification of a system) entails a global perspective of the entire system under consideration. There are not yet widely accepted methods of factoring the various aspects of a trust assessment and then reassembling the results into a statement about the whole.

These conflicting perspectives of local production and global evaluation analysis are particularly evident in the field of database management, but they are by no means limited to that field. Thus the publication of this Interpretation requires consideration of how to deal with systems built in parts in the absence of a general treatment of the topic. On the other hand, any treatment of the issue in the context of trusted database management will have significant influence in other fields where the same or similar problems arise, just because of community review and NCSC publication. The approach taken in this document is to address the issues of evaluating systems built of parts in a way that is independent of the field of trusted database management. This conscious attitude of generality is intended to make clear the distinction between the larger system-of-parts issues and the more specific DBMS issues.

PART 1, "TECHNICAL CONTEXT," is divided into six sections. Section TC-2, "Reference Monitor Perspective," summarizes the role of the reference monitor concept in both the TCSEC and the assessing of a system for its trust characteristics. Section TC-3, "Need for Evaluation by Parts," deals with the need to extend the reference monitor perspective slightly to begin to address the evaluation of systems constructed of separate parts. Section TC-4, "TCB Subsets," is the heart of PART 1. That section introduces a conservative extension to the reference validation mechanism, TCB subsets. This extension provides the basis for being able to undertake evaluation of systems built in parts in a way that allows re-use of the results of separate evaluations (whether those evaluations were performed before the current evaluation was begun or whether the separate evaluations overlap in time). Section TC-5, "General Interpreted Requirements," outlines the application of the TCSEC requirements to individual TCB subsets when an evaluation by parts is being done. Section TC-6, "Design Choices" describes the general process of applying TCB subsets and meeting the conditions for evaluation by parts. The treatment in this section is general and not limited to DBMSs; DBMS-specific issues are discussed in the appendices.

TC-2 REFERENCE MONITOR PERSPECTIVE

Building or evaluating a system for compliance with the requirements of a particular class in the TCSEC is based on the perspective of the Trusted Computing Base (TCB). The notion of the TCB is central to the entire concept of assessing systems for trust. The reference monitor described in the Anderson report [1] is the basis of the notion of a TCB, as described in the TCSEC:

> For convenience, these evaluation criteria use the term Trusted Computing Base to refer to the reference validation mechanism, be it a security kernel, front-end security filter, or the entire trusted computer system. [8, p. 67]

Even in those lower classes (below B2) where the reference monitor concept and reference validation mechanisms are not mentioned explicitly, the perspective of the reference monitor and its implementation as a reference validation mechanism is present. Specifically, there are requirements for (1) identifying the policy being enforced, (2) identifying subjects and objects, (3) providing evidence that the operation of the reference validation mechanism matches the high-level description of the user interface, and (4) demonstrating isolation of the TCB.

Therefore, all TCSEC evaluations share the initial conceptual steps of identifying the mediation policy, the subjects, and the objects. Starting from a global system perspective, the initial step is to identify the access mediation policy that will be enforced. One must then identify those active system entities that are candidates for being the "subjects" whose access to objects the TCB will mediate. Similarly, one must identify those passive entities, those data repositories, that are candidates for being the "objects," access to which the TCB will mediate.

As usual, the existence of an abstraction within a system does not make it necessarily a reference-monitor object, i.e., one visible at the TCB interface. This is because the TCB will make use of system abstractions for both its internal processes and its storage requirements. Those entities, while being stored in system "objects," will not be available to untrusted processes (that is, they are not exported by the TCB). Thus the analytical, iterative step is the separation of candidate subjects and objects into those that are mediated by the TCB and those that are not.

The various trust classes include requirements, at varying levels of completeness and rigor, that the basic reference monitor perspective of mediating access of subjects to objects be implemented in a correct, self-protecting, and non-bypassable manner. The core requirements of the TCSEC classes focus on these reference monitor imperatives. The increasingly strict requirements for visibility into the system design and implementation (structure, documentation, testing, configuration, and distribution requirements) all support the notion of checking the system's conformance to its identified intent with regard to the subjects, objects, and policy.

TC-3 NEED FOR EVALUATION BY PARTS

The need to be able to evaluate and certify systems built in parts is increasingly evident. This need is seen in a variety of contexts:

The need to evaluate and certify systems built out of parts sold by different vendors, a situation especially prevalent in the field of trusted DBMSs.

The need to re-assess systems that have undergone either small- or large-scale improvements and are already evaluated and placed on the Evaluated Products List (EPL).

The general problem of "families of systems," systems that exist on a spectrum of hardware bases or that can be customized for a user's specific needs.

In all such cases, two related versions of a system are largely similar. It should be possible to undertake evaluations and certifications in such a way that the entire assessment does not have to be re-done for every slight variation that appears. The current state of technology, however, places limitations on what can be accomplished in this regard; it is not currently possible to determine the trust characteristics of a system on the basis of an arbitrary collection of subparts. The overall task of trust assessment entails so many interrelated subtasks that the ability to separate and reassemble is not well understood.

In this circumstance of needing to be able to accommodate evaluation of a system built in parts and the lack of consensus about how this can be done in a technically sound fashion, a conservative approach must be adopted. The following are required: (1) a clear description of what "parts" will be considered for separate evaluation; (2) a clear description of the conditions under which such an evaluation by parts will be undertaken; and (3) a general interpretation of TCSEC requirements as they would apply when a system is being evaluated by parts. The "parts" that will be considered by separate evaluation are called "TCB subsets," the topic of Section TC-4 below.

TC-4 TCB SUBSETS

TC-4.1 INTRODUCTION

To attempt an evaluation of a TCB by splitting it into parts, there must be available a precise definition of what parts are candidates for separate evaluation (that is, for evaluation by parts) as well as any other conditions that must be satisfied before an evaluation by parts will be undertaken. This section defines "TCB subset" as a strict and conservative extension of the traditional concept of the reference validation mechanism (RVM) as a method of delineating which parts of a TCB can be candidates for separate evaluation. The definition of TCB subsets, as well as explanatory and motivational material, is included below in Section TC-4.2, "TCB Subsets Context." Section TC-4.3 addresses the conditions that must be satisfied by a TCB divided into a set of TCB subsets before evaluation by parts will be undertaken. These conditions assure that the structure of and relationships among TCB subsets will satisfy TCSEC requirements, especially those derived from the reference monitor concept.

TC-4.2 TCB SUBSETS CONTEXT

TC-4.2.1 DEFINITION OF TCB SUBSETS

A TCB subset M is a set of software, firmware, and hardware (where any of these three could be absent) that mediates the access of a set S of subjects to a set O of objects on the basis of a stated access control policy P and satisfies the properties:

1) M mediates every access to objects in O by subjects in S;
2) M is tamper resistant; and
3) M is small enough to be subject to analysis and tests, the completeness of which can be assured.

M uses resources provided by an explicit set of more primitive TCB subsets to create the objects of O, create and manage its data structures, and enforce the policy P. If there are no TCB subsets more primitive than M, then M uses only hardware resources to instantiate its objects, to create and manage its own data structures, and to enforce its policy.

The above definition does not explicitly prohibit an access control policy P that allows trusted subjects. The implications for the evaluation process when a TCB subset's policy allows or does not allow such trusted subjects are substantial and are discussed in Section TC-6.4. As described in TC-4.3, one of the conditions for an evaluation by parts of a TCB made up of TCB subsets is that all the trusted subjects of each TCB subset be included in that TCB subset.

TC-4.2.2 MOTIVATION

The definition of "TCB subset" is intended to parallel the definitions of the reference monitor and reference validation mechanism found in the Anderson report [1] and included in the TCSEC itself. "The term Trusted Computing Base [refers] to the reference validation mechanism, be it security kernel, front-end security filter, or the entire trusted computer system." [8, p. 67] "TCB subset" is defined exactly like a reference validation mechanism, with only one difference, that it does not necessarily extend to the hardware. Rather, a TCB subset uses either hardware resources or the resources provided by other, more primitive TCB subsets. Thus TCB subsets build on abstract machines, either physical hardware machines or other TCB subsets. Just like reference validation mechanisms, a TCB subset must enforce a defined access control policy.

TC-4.3 CONDITIONS FOR EVALUATION BY PARTS

Building or evaluating a system using the definition of TCB subsets in the section above requires meeting six conditions in addition to demonstrating that the candidate TCB subsets satisfy the properties appropriate to the evaluation target class. The conditions are as follows:

The candidate TCB subsets are identified;

The system policy is allocated to the candidate TCB subsets;

Each candidate TCB subset M[i] includes all the trusted subjects with respect to its technical policies P[i];

The TCB subset structure or architecture is explicitly described;

Each TCB subset occupies distinct subset-domains; and

The more primitive TCB subsets provide support for the RVM arguments for less primitve TCB subsets.

These conditions are described below.

TC-4.3.1 CANDIDATE TCB SUBSETS

The first condition is that the relevant elements of each candidate TCB subset M[i] be identified. The hardware, firmware, and software which compose the TCB subset need to be clearly identified, along with the subjects and objects. In terms of Section TC-4.2.1, this condition is the identification of M[i], S[i], and O[i]. There may be no objects mediated by more than one TCB subset. That is, there cannot be an object O that is both in O[i] and O[j].

TC-4.3.2 POLICY ALLOCATION

The next condition is policy allocation, the description of the technical policy P[i] for each identified M[i] along with the relation of these policies to the system policy P. The policies P[i] will be expressed in terms of subjects in S[i] and objects in O[i]. Thus, to satisfy the TCSEC requirement that the (composite) TCB enforce its stated policy P, each rule in P must be traceable through the structure of the candidate TCB subsets to the TCB subset(s) where that enforcement occurs. See Sections TC-5.2.1.1 and TC-5.2.1.4.

TC-4.3.3 TRUSTED SUBJECTS INCLUDED

Every trusted subject with respect to P[i] must be part of the TCB subset M[i]. This condition makes possible separate evaluation of TCB subsets with respect to "local" requirements. When this condition is not met, evaluation of candidate TCB subsets cannot be guaranteed on an evaluation by parts basis. This situation is treated in Section 6.4.

TC-4.3.4 TCB SUBSET STRUCTURE

The TCB subsets will exhibit a structure based on the ordering implied by dependency. TCB subset A is less primitive than TCB subset B if (a) A directly depends on B or (b) a chain of TCB subsets from A to B exists such that each element of the chain directly depends on its successor in the chain. In this case we use the phrase "TCB subset B is more primitive than TCB subset A" synonymously.

The sense of "directly depend" in (a) is exactly the following: it is not possible to verify the implementation of A with respect to its specification without a statement about the specification of B.

More precisely, for a candidate TCB subset M, let sM denote the specification of M. The specification will include, as a minimum, the statement of the technical policy P of M. Further, let vM denote the (engineering) demonstrations of the correct implementation of M with respect to its specification. A (candidate) TCB subset A "depends (for its correctness)" on (candidate) TCB subset B if and only if the arguments of vA assume, wholly or in part, that sB has been implemented correctly. (See Appendix B, item 5 for additional discussion.)

The proposed structure of TCB subsets has to be identified. The order must be a partial order. (Without a partial order, there could be circular chains of dependencies that would inhibit separate evaluations of the TCB subsets.)

TC-4.3.5 SEPARATE SUBSET-DOMAINS

The candidate TCB subsets must operate in near isolation from each other, with the only interaction between them being that explicitly asserted as part of the interface. In the case of reference monitors, many existing implementations have relied on the domain concept [23] which supports the assertions of non-bypassability and self protection. A natural extension of the domain concept will be required for isolation of TCB subsets from each other and from non-TCB entities.

A subset-domain is a set of system domains. Each candidate TCB subset must occupy a distinct subset-domain such that modify-access to a TCB subset's subset-domain is permitted only to that TCB subset and (possibly) to more primitive TCB subsets. This requirement ensures that the structure of subset-domains with respect to access is consonant with the dependency relation among TCB subsets.

TC-4.3.6 SUPPORT FOR RVM ARGUMENTS

Candidate TCB subsets must satisfy the three RVM properties included in the definition in TC-4.2.1 in order to complete evaluation by parts successfully. TCB subsets that build on resources and functionality provided by more primitive TCB subsets must rely on assured and evaluatable characteristics of those more primitive TCB subsets. A convincing argument must be advanced that the features, characteristics, and assurances provided by the more primitive TCB subsets are capable of supporting RVM arguments for the less primitive TCB subsets.

The first property (mediating every access) requires that there is no way of bypassing the mediation provided by TCB subset M for its objects, either directly or by unexpected side-effects of interactions with other TCB subsets. A variety of approaches could suffice for demonstrating this property.

The second property (tamper resistance) requires that the policy-critical code and data for the less primitive TCB subset be protected from any alteration not specifically allowed by the TCB subset. A variety of approaches could suffice for demonstrating this property.

The third property (completeness of testing and analysis for correctness) requires the (engineering) demonstrations vM[i] of the correctness of each less primitive candidate TCB subset M[i]. A convincing argument must therefore be advanced that the specifications sM[k] for all of the more primitive TCB subsets M[k] on which M[i] depends will suffice for establishing vM[i].

TC-4.4 EVALUATION ALTERNATIVES

As noted earlier, the need to evaluate systems whose elements are developed separately, possibly by independent developers, results in the need to define TCB subsets. That is not to say, however, that design by TCB subsetting and the subsequent evaluation by parts are the only alternatives. Rather, there are three distinct possibilities.

A system TCB, regardless of any internal structure, may be viewed as a single entity. In such a case, the evaluation may proceed essentially as an evaluation against the TCSEC. This case is examined in Section TC-6.2.

A system TCB may be presented as a subsetted architecture composed of a number of candidate TCB subsets. Given that each of the candidate TCB subsets satisfies the conditions set forth in Section TC-4.3, an evaluation by parts is possible. This case is described in Section TC-6.3.

It may be possible to satisfy only some of the conditions for evaluation by parts. This situation might arise when a previously evaluated TCB or TCB subset is modified to accommodate the policy-enforcing elements of a new application layer. A special case of this situation is described in Section TC-6.4. In such cases, depending upon the particulars, it may be possible to realize some of the savings in evaluation effort. However, no general statements can be made for these cases.

TC-5 GENERAL INTERPRETED REQUIREMENTS

TC-5.1 OVERVIEW

This section provides specific interpretations of those TCSEC requirements that are particularly relevant for subsetted architectures and evaluation by parts. Its organization is derived from the structure of the TCSEC requirements. For each relevant TCSEC requirement there is a discussion of how that requirement is interpreted in an evaluation by parts.

For conciseness, only the requirements headings applicable for A1 systems are included below. Thus, the interpretation guidance below should be applied only as demanded by the requirements for the target class of the candidate trusted system. For a system targeted at an evaluation class below A1, only those requirements that pertain to the target class apply to the TCB subsets making up that system.

A listing of the requirements and interpretations by TCSEC class is presented in Appendix A. The rationale for the applicability of the TCSEC requirements to TCB subsets alone or to the TCB as an entirety is described in Appendix B, item 7.

TC-5.2 DETAILED REQUIREMENTS

TC-5.2.1 SECURITY POLICY

TC-5.2.1.1 Discretionary Access Control

The discretionary access control requirements apply as stated in the TCSEC to every TCB subset whose policy includes discretionary access control of its subjects to its objects. Any TCB subset whose policy does not include such discretionary access control is exempt from this requirement.

TC-5.2.1.2 Object Reuse

This requirement applies as stated in the TCSEC to every TCB subset in the TCB.

TC-5.2.1.3 Labels

This requirement applies as stated in the TCSEC to every TCB subset whose policy includes mandatory access control of its subjects to its objects. Any TCB subset whose policy does not include such mandatory access control is exempt from this requirement.

Label Integrity

This requirement applies as stated in the TCSEC to every TCB subset whose policy includes mandatory access control of its subjects to its objects. Any TCB subset whose policy does not include such mandatory access control is exempt from this requirement.

Exportation of Labeled Information

This requirement applies as stated in the TCSEC to every TCB subset whose policy includes mandatory access control of its subjects to its objects. Any TCB subset whose policy does not include such mandatory access control is exempt from this requirement.

Subject Sensitivity Labels

This requirement applies as stated in the TCSEC to the entire TCB. The cooperative action of the TCB subsets making up the TCB must satisfy the requirement.

Device Labels

This requirement applies as stated in the TCSEC to every TCB subset whose policy includes mandatory access control of its subjects to its objects and has attached physical or virtual devices. Any TCB subset whose policy does not include such mandatory access control or has no attached physical or virtual devices is exempt from this requirement. This requirement can be satisifed by the cooperative action of more than one TCB subset.

TC-5.2.1.4 Mandatory Access Control

This requirement applies as stated in the TCSEC to every TCB subset whose policy includes mandatory access control of its subjects to its objects. Any TCB subset whose policy does not include such mandatory access control is exempt from this requirement.

TC-5.2.2 ACCOUNTABILITY

TC-5.2.2.1 Identification and Authentication

This requirement applies as stated in the TCSEC to the entire TCB. The cooperative action of the TCB subsets making up the TCB must satisfy the requirement.

If the TCB is composed of TCB subsets, one TCB subset may either rely on a mechanism in another TCB subset to provide identification and authentication services or provide the services directly. Similarly, that TCB subset may rely on a mechanism in another more primitive TCB subset to ensure that the security level of subjects external to the TCB that may be created to act on behalf of the individual user are dominated by the clearance and authorization of that user. Each TCB subset may maintain its own identification and authentication data or one central repository may be maintained. If each TCB subset has its own data, then the information for each individual user must be consistent among all subsets.

Trusted Path

This requirement applies as stated in the TCSEC to the entire TCB. The cooperative action of the TCB subsets making up the TCB must satisfy the requirement.

When TCB subsets are used, the requirement for trusted path at levels B2 and above remains applicable to the entire TCB. The need for trusted path "when positive TCB-to-user connection is required (e.g., login, change subject security level)" can require user interaction with virtually any TCB subset within the TCB. The implementation of trusted path could be localized in a single TCB subset. Alternatively, it could be implemented in more than one TCB subset if the separate implementations together comply with the system security policy.

TC-5.2.2.2 Audit

This requirement applies as stated in the TCSEC to the entire TCB. The cooperative action of the TCB subsets making up the TCB must satisfy the requirement.

A TCB subset may maintain its own security audit log, distinct from that maintained by more primitive TCB subsets, or it may use an audit interface provided by a different TCB subset allowing the audit records it generates to be processed by that TCB subset.

If the TCB subset uses different user identifications than a more primitive TCB subset, there shall be a means to associate audit records generated by different TCB subsets for the same individual with each other, either at the time they are generated or later.

Any TCB subset wherein events may occur that require notification of the security administrator shall be able to do the following: (1) detect the occurrence of these events, (2) initiate the recording of the audit trail entry, and (3) initiate the notification of the security administrator. The ability to terminate events (2) and (3) above may be provided either in the TCB subset within which they occur, or in the TCB subset(s) where actions that lead to the event were initiated.

The monitoring and notification requirements may require cooperation between multiple distinct TCB subsets or multiple instantiations of the same TCB subset. For example, to detect the accumulation of events for a single user it may be necessary to collect events from several subjects in distinct processes that are surrogates for the same user. As another example, there may be a single TCB subset that collects events from a number of other TCB subset instantiations and, based on its analysis of them, notifies the security administrator.

TC-5.2.3 ASSURANCE

TC-5.2.3.1 Operational Assurance

System Architecture

This requirement applies as stated in the TCSEC to every TCB subset, with the following additional interpretations.

The TCB must provide domains for execution that are allocated to and used by TCB subsets according to the subset-domain condition for evaluation by parts. A most primitive TCB subset must provide domains for execution. A less primitive TCB subset must make use of domains provided by a more primitive TCB subset. A less primitive TCB subset may provide further execution domains but is not required to do so.

Similarly, the TCB must provide distinct address spaces for untrusted processes. A most primitive TCB subset must provide distinct address spaces for its subjects. A less primitive TCB subset must make use of the distinct address space provided by a more primitive TCB subset. A less primitive TCB subset may provide more fine-grained distinct address spaces, but is not required to do so.

In general, requirements specifically referring to hardware or firmware apply only to TCB subsets that include hardware or firmware. The exception is the requirement that the TCB make effective use of available hardware. This requirement applies to those TCB subsets that use resources provided by more primitive TCB subsets in lieu of hardware. Those TCB subsets are required to make effective use of the protection-relevant features exported to it by the more primitive TCB subsets (e.g., execution domains, support for distinct address spaces) to separate those elements that are protection-critical from those that are not.

System Integrity

This requirement applies as stated in the TCSEC to every TCB subset that includes hardware or firmware. Any TCB subset that does not include hardware or firmware is exempt from this requirement.

Covert Channel Analysis

This requirement applies as stated in the TCSEC to the entire TCB. When the TCB is made up entirely of TCB subsets meeting the conditions for evaluation by parts, analysis of the individual TCB subsets satisfies this requirement. Otherwise, covert channel analysis of the entire TCB must be performed (even if the results of covert channel analysis of the individual TCB subsets were available).

Trusted Facility Management

This requirement applies as stated in the TCSEC to the entire TCB. Any "operator" or "administrator" functions intrinsic to an individual TCB subset must be supported by that TCB subset or by a more primitive TCB subset.

Trusted Recovery

This requirement applies as stated in the

TCSEC to the entire TCB and to the individual TCB subsets. The cooperative recovery actions of the TCB subsets making up the TCB must provide trusted recovery for the overall TCB. Otherwise, trusted recovery evaluation must address the entire TCB (even if the individual TCB subsets meet the trusted recovery requirements).

TC-5.2.3.2 Life-Cycle Assurance

Security Testing

This requirement applies as stated in the TCSEC to the entire TCB. If a TCB consists of TCB subsets meeting the conditions for evaluation by parts, the satisfaction of the requirements by each TCB subset satisfies the requirement for the entire TCB. Otherwise, security testing of the entire TCB must be performed (even if the results of testing the individual TCB subsets were available).

Design Specification and Verification

This requirement applies as stated in the TCSEC to every TCB subset, with the following specific additional interpretations.

It must be demonstrated that the security policy enforced by the composite TCB is represented by the collection of policy models for the individual TCB subsets.

The argument that the descriptive top level specification (DTLS) and formal top level specification (FTLS) are consistent with the TCB interface applies to the entire TCB. There is required an explicit and convincing description of how the set of top level specifications (one for each TCB subset) describes the TCB interface in terms of exceptions, errors, and effects.

Configuration Management

This requirement applies as stated in the TCSEC to every TCB subset in the TCB, with the following additional interpretation.

Because subsets of the TCB may be developed independently, a single configuration management system may not be feasible. However, the combination of configuration management systems used to support all the TCB subsets must meet all the stated requirements. The information describing the interrelations between separate TCB subsets and separate security policy models falls into the category of "all documentation and code associated with the current version of the TCB" in the TCSEC requirements.

Trusted Distribution

This requirement applies as stated in the TCSEC to the entire TCB. It can be met by satisfying the requirements for each TCB subset if procedures exist for assuring that all TCB subsets upon which a particular TCB subset depends (that is, the more primitive TCB subsets) are exactly the same version as specified for the TCB subset in question.

TC-5.2.4 DOCUMENTATION

TC-5.2.4.1 Security Features User's Guide

This requirement applies as stated in the TCSEC to every TCB subset in the TCB. This collection of guides must include descriptions of every TCB subset in the TCB and explicit cross-references to other related user's guides to other TCB subsets, as required. In addition, interactions between mechanisms within different TCB subsets must be clearly described.

TC-5.2.4.2 Trusted Facility Manual

This requirement applies as stated in the TCSEC to the TCB and to every TCB subset in the TCB.

This requirement can be met by providing a set of manuals, one for each distinct (non-replicated) TCB subset. Each manual shall address the functions and privileges to be controlled for the associated TCB subset. Additionally, it must clearly show the interfaces to, and the interaction with, more primitive TCB subsets. The manual for each TCB subset shall identify the functions and privileges of the TCB subsets on which the associated TCB subset depends. Also, the TCB subset's manual shall identify which, if any, configuration options of the more primitive TCB subsets are to be used for the composite TCB to operate securely.

Any pre-defined roles supported (e.g., database administrator) by the TCB subset shall be addressed.

The means for correlating multiple audit logs and synonymous user identifications from multiple TCB subsets (if such exist) shall also be addressed.

The trusted facility manual shall describe the composite TCB so that the interactions among the TCB subsets can be readily determined. Other manuals may be referenced in this determination. The manuals that address the distinct modules of the TCB and the generation of the TCB need to be integrated with the other trusted facility manuals only to the extent that they are affected by the use and operation (versus the development) of the composite TCB.

The TCB modules that contain the reference validation mechanism must be associated with the TCB subset to which they belong. The procedure for generating a new TCB after modifying only one (or several) TCB subsets must be described. This may be accommodated by independent regeneration of the individual TCB subsets or by regeneration of only the affected TCB subsets.

TC-5.2.4.3 Test Documentation

This requirement applies as stated in the TCSEC to the composite TCB.

TC-5.2.4.4 Design Documentation

This requirement applies as stated in the TCSEC to the composite TCB, with the following specific additional interpretations:

Requirements concerning models, FTLS and DTLS, apply to individual TCB subsets.

The requirement concerning the description of interfaces between modules of the TCB includes the interfaces between TCB subsets.

The documentation of the implementation of a reference validation mechanism must include justification for meeting the conditions for evaluation by parts.

The A1 requirement to describe clearly non-FTLS internals of the TCB applies to TCB subsets.

TC-5.3 SUMMARY OF THE REQUIREMENTS

The requirements interpretations in Section

TC-5.2 above can be grouped into two categories: those that apply to individual TCB subsets and those that apply totally or in part to the overall TCB. For purposes of exposition, the former category will be termed "local requirements," that is, those for which separate analysis of the individual TCB subsets suffices to determine compliance for the composite TCB. The latter are termed "global requirements," that is, those which require analysis of the entire system and for which separate analysis of the individual TCB subsets does not suffice.

TC-5.3.1 LOCAL REQUIREMENTS

Discretionary Access Control;
Object Reuse;
Labels (except Subject Sensitivity Labels);
Mandatory Access Control;
System Architecture (except domains for
  execution and distinct address spaces);
System Integrity;
Configuration Management;
Security Features User's Guide;
Design Documentation
models,
DTLSs,
FTLSs, and
non-FTLS internals.

TC-5.3.2 GLOBAL REQUIREMENTS

Subject Sensitivity Labels;
Identification and Authentication;
Trusted Path;
Audit;
System Architecture domains for execution, and distinct
 address spaces;
Covert Channel Analysis;
Trusted Facility Management;
Trusted Recovery (also applies to
 individual TCB subsets);
Security Testing;
Design
Specification and Verification
correspondence between system
 policy and the set of TCB subset models
consistency of TCB interface with the
 set of TCB subset DTLSs, and
consistency ofTCB interface with the
 set of TCB subset FTLSs;
Trusted Distribution;
Trusted Facility Manual (also applies to individual TCB
 subsets);
Test Documentation; and

Design Documentation (except models, DTLSs, FTLSs, and non-FTLS internals).

TC-6 DESIGN CHOICE

TC-6.1 OVERVIEW

This section examines the several design choices available for constructing systems in parts and the consequences of each when attempting to perform an evaluation by parts. The first case described is that of a TCB evaluated under the TCSEC without benefit of TCB subset structuring. This case is of value for several reasons: it serves as a reference point; it can be considered the degenerate case of subsetting; and it is always an option to evaluate any TCB, regardless of internal structure, as a monolith. The second and third cases are presented in terms of a configuration of exactly two subsets; the generalization to more than two TCB subsets is straightforward. The second case is that of a subsetted architecture that exactly satisfies the conditions for evaluation by parts. The third case represents a special case of an altered TCB, one which is implemented using trusted subjects.

Note that no evaluation using TCB subsets and evaluation by parts results in a TCB subset receiving an evaluation rating. Rather, the entire system, with its composite TCB, is evaluated and receives a rating. However, evaluation by parts is intended to allow the results of local analysis of individual TCB subsets to be distinguishable and separately referencable. For further discussion of this topic, see Appendix B, item 10.

TC-6.2 A SINGLE TCB SUBSET

The evaluation of a TCB consisting of a single TCB subset is equivalent to a straightforward evaluation against the TCSEC. The conditions for evaluation by parts (Section TC-4.3) reduce to requirements found in the TCSEC itself.

TC-6.2.1 ANALYSIS OF THE CONDITIONS

TC-6.2.1.1 Condition 1: Candidate TCB Subsets

The TCB (hardware, software, and firmware), as distinguished from the rest of the computing system, must be identified. This is a basic requirement for TCSEC evaluation. Similarly, the subjects and objects within the system must be identified. The requirement that no more than one TCB subset mediate access to any particular object is satisfied, because there is only one TCB subset.

TC-6.2.1.2 Condition 2: Policy Allocation

The policy P enforced by the TCB (subset) must be identified. The demonstration that the TCB (subset) enforces that policy will be a description of how the TCB performs access mediation between the system's subjects and objects according a system-level description of limitations on access (the technical policy P[i] of the definition). The tracing of the policy to the system design and behavior is part of the stated TCSEC requirements.

TC-6.2.1.3 Condition 3: Trusted Subjects Included

This condition is satisfied in the same manner as it is in evaluations under the TCSEC. Specifically, the TCB boundary is shown to be the interface that is presented to untrusted subjects.

TC-6.2.1.4 Condition 4: TCB Subset Structure

Satisfaction of this condition (TC-4.3.4) is immediate, because there is only one TCB subset.

TC-6.2.1.5 Condition 5: Separate Subset-Domains

Satisfaction of the separate subset-domain condition (TC-4.3.5) is identical to the C1 through A1 requirement that "the TCB maintain a domain for its own execution that protects it from external interference or tampering." [8, p. 13 et al.]

TC-6.2.1.6 Condition 6: Support for RVM Arguments

Satisfaction of this condition (TC-4.3.6) is immediate, inasmuch as there are no less primitive TCB subsets that must be demonstrated to satisfy RVM requirements.

TC-6.2.2 EVALUATION CONSEQUENCES

In this case, the evaluation of the (single) TCB subset proceeds exactly like an evaluation under the TCSEC. Demonstration that the candidate system meets all the global and local requirements (as they apply to the target evaluation class) includes the consideration of each requirement as it applies system's philosophy of protection, design, documentation, and implementation. The system must be shown to exhibit the properties of a reference validation mechanism, appropriate to the target class.

TC-6.3 TWO TCB SUBSETS, MEETING THE CONDITIONS

This case is of a TCB that consists of two candidate TCB subsets, A and B, where A is the most primitive TCB subset. That is, B uses the abstractions provided by A (the objects, in particular) as its resource for the construction and exportation of its own abstractions. B also uses the abstractions provided by A for its metadata (that is, internal data structures) that make it possible for B to instantiate its exported abstractions as well as keep records that enable it to correctly enforce its stated policy. In terms of the discussion of Section TC-4.3.4, TCB subset B directly depends on TCB subset A. It will be assumed that TCB subset A enforces mandatory and discretionary policies on its objects and that TCB subset B enforces a discretionary policy on the objects it exports. Additionally, all trusted subjects of A are contained within A. Thus, every subject of A (including all the active entities that make up the logic of B) operates at a single sensitivity level. It will further be assumed for th is example that the mechanisms for domains and thus for subset-domains are independent of the mandatory and discretionary access control policy enforcement mechanisms.

TC-6.3.1 ANALYSIS OF THE CONDITIONS

TC-6.3.1.1 Condition 1: Candidate TCB Subsets

The TCB (hardware, software, and firmware), as distinguished from the rest of the computing system, must be identified. This is a basic requirement for TCSEC evaluation. Similarly, the subjects and objects within the system must be identified.

In this case, all the hardware, software, and firmware that make up the TCB must be identified as being part of either TCB subset A or TCB subset B. The subjects and objects mediated by A (call them the "A-subjects" and "A-objects" for this discussion) must be identified. Similarly the B-subjects and B-objects must also be identified.

The additional requirement in Section TC-4.3.1 that "there may not be any objects mediated by more than one TCB subset" means that there can be no B-object that is also an A-object.

TC-6.3.1.2 Condition 2: Policy Allocation

The policy P enforced by the whole TCB must be identified. In addition, the policy enforced by A (call it the A-policy), stated in terms of the A-subjects and the A-objects, must be identified. Similarly, the B-policy, stated in terms of the B-subjects and the B-objects, must be identified.

TC-6.3.1.3 Condition 3: Trusted Subjects Included

As was stated above, TCB Subset A contains all its own trusted subjects. There may be trusted subjects with respect to the policy of A, but all such subjects execute in the subset-domain of A.

TC-6.3.1.4 Condition 4: TCB Subset Structure

Because B directly uses the A-objects and its logic is embodied in A-subjects, the structure of the TCB subsets is precisely "TCB subset A is more primitive than TCB subset B." This is a partial order.

TC-6.3.1.5 Condition 5: Separate Subset-Domains

Satisfaction of the separate subset-domain condition requires that a set of domains provided by the system be identified as being the domains "occupied" by A and B. The domain, or domains, occupied by A is exactly the "domain for its own execution" found in the TCSEC requirements. The domain or domains occupied by TCB subset B must not be modifiable by any code or other system entity except possibly by TCB subset A.

TC-6.3.1.6 Condition 6: Support for RVM Arguments

Satisfying the condition for RVM arguments requires demonstrating the plausibility of being able to establish the three requisite properties of an RVM. The first property requires that no B-subject be allowed to access B-objects without those accesses being mediated by TCB subset B. The tamper resistance property requires that TCB subset A provide a way that TCB subset B can be designed and implemented such that A-subjects that are not part of B's implementation cannot tamper with B's policy-critical code or data. The third RVM property must be satisfied by the individual TCB subsets. The degree to which each TCB subset must satisfy this property is commensurate with the evaluation class of the TCB.

TC-6.3.2 EVALUATION CONSEQUENCES

In this case, the evaluation of the two TCB subsets requires that each meet TCSEC requirements applicable to each TCB subset viewed individually and that the two TCB subsets combine in a way to meet all the TCSEC requirements stated for the target class.

All local requirements are imposed on the two TCB subsets, A and B, individually. If each TCB subset can meet the requirements of the target class, viewed as if it were a separate TCB, the only areas where additional evaluation or accreditation work might be required are those areas where the sum of the analysis of the parts is not necessarily complete and convincing. Those areas requiring additional work are exactly the set of global requirements described in Section TC-5.3.2.

Demonstrating that the candidate system meets the TCSEC requirements (as they apply to the target evaluation class) requires that both A and B be evaluated with respect to the local requirements of the target class and that the composite TCB be evaluated for global requirements. In this case, full testing of TCB subset A against all the requirements (both local and global) simplifies the task of demonstrating satisfaction of the global requirements, both for B and for the entire TCB.

Suppose, for example, that TCB subset A has been subjected to security testing appropriate to the target class and has been shown to be adequately resistant to penetration attacks. This means that within the confidence level provided by the testing requirement, no A-subject can subvert A's enforcement of its policy. In this situation, every active entity in B is an A-subject and hence B can neither penetrate A nor be induced to do so by any B-subject. Thus, no further testing of A will be required to determine whether the entire TCB is resistant to penetration; any additional penetration testing can be limited to determining the ability of B to withstand assault.

Similarly, if A has been searched for covert channels (as required for its target class requirements), then no further search for covert channels will be required, either in the analysis of B or in the overall consideration of the entire TCB. Note that if B implements a mandatory access control policy (e.g., integrity), then it would be necessary to perform a covert channel analysis on B, but no further covert channel analysis of A would be required.

The ability of users to determine the current sensitivity level of B-subjects operating on their behalf will have to be shown by considering the TCB subsets A and B together. This requirement is satisfied immediately if the argument relies exclusively on A meeting the requirement.

TC-6.4 TWO TCB SUBSETS, NOT MEETING THE CONDITIONS

This case also concerns a TCB that consists of two candidate TCB subsets, C and D. C is the most primitive TCB subset. That is, D uses the abstractions provided by C (the objects, in particular) as its resource for the construction and exportation of its own abstractions. D also uses the abstractions provided by C for its metadata (that is, internal data structures) that make it possible for D to instantiate its exported abstractions as well as keep records that enable it to correctly enforce its stated policy. In terms of the discussion of Section TC-4.3.4, TCB subset D directly depends on TCB subset C. Additionally, D is trusted with respect to C. That is, some of the C-subjects which make up TCB subset D execute as trusted processes of C. Here also, as in the previous example, it is assumed that C implements mandatory and discretionary policies over its objects. Further, the intent of D is to implement a discretionary policy over the objects it exports. However, because D includes subjects which are trusted relative to C's policy demonstration of the full and correct enforcement of the mandatory policy requires analysis of both C and D and is no longer localized to TCB subset C. It will be assumed that the mechanisms for domains and thus for subset-domains are independent of the mandatory and discretionary access control policy enforcement mechanisms.

This case can be viewed as a special case of a previously evaluated TCB which has been altered. However, the alteration takes the form of a less primitive subset which is implemented, at least in part, with trusted subjects (i.e., some of the

C-subjects are trusted subjects which execute in the subset-domain of D). Although this case may appear, intuitively, to be different from that of arbitrary alteration of a previously evaluated TCB, the example demonstrates that such an approach makes it impossible to perform an evaluation by parts.

TC-6.4.1 ANALYSIS OF THE CONDITIONS

TC-6.4.1.1 Condition 1: Candidate TCB Subsets

The identification of the TCB (hardware, software, and firmware) as distinguished from the rest of the computing system is a basic requirement for TCSEC evaluation. Likewise, the subjects and objects within the system must be identified.

In this case, all the hardware, software, and firmware that make up the TCB must be identified as being part of either TCB subset C or TCB subset D. The C-subjects and C-objects mediated by C have to be identified. Similarly the D-subjects and D-objects must also be identified.

The additional requirement in Section TC-4.3.1 that "there may not be any objects mediated by more than one TCB subset" means there can be no D-object that is also a C-object.

TC-6.4.1.2 Condition 2: Policy Allocation

The policy P enforced by the whole TCB must be identified. In addition, the individual policy enforced by C (call it the C-policy) must be identified in terms of the C-subjects and the C-objects. Similarly, the D-policy, stated in terms of the D-subjects and the D-objects, must be stated. In this case, the C-policy will include the strict enforcement of a mandatory access control policy that allows trusted subjects to execute in the subset-domains which compose TCB subset D.

TC-6.4.1.3 Condition 3: Trusted Subjects Included

This condition is not satisfied because D includes at least one subject that is trusted with respect to C. Hence a subject that is trusted with respect to the policy of C is outside C, and evaluation by parts is not an option. If TCB subset C had previously been evaluated, then this is an example of an altered TCB, albeit a special case. The change consists of the addition of one or more trusted C-subjects in D whose effect on the behavior of C cannot be predicted a priori. An assessment of the impact of D on the behavior of C cannot be made strictly by an examination of the trusted subjects and the definition of C's interface. A global assessment of C and D is required.

TC-6.4.1.4 Condition 4: TCB Subset Structure

Because D directly uses the C-objects and its logic is embodied in C-subjects, the structure of the TCB subsets is precisely "TCB subset C is more primitive than TCB subset D." This is a partial order.

TC-6.4.1.5 Condition 5: Separate Subset-Domains

Satisfying the separate subset-domain condition (TC-4.3.5) requires identifying the set of system domains (likely administered by the most primitive TCB subset C) as the domains "occupied" by C and D. The domain, or domains, occupied by C is exactly the "domain for its own execution" found in the TCSEC requirements. The domain or domains occupied by TCB subset D must not be modifiable by any code or other system entity except possibly by a part of TCB subset C.

TC-6.4.1.6 Condition 6: Support for RVM Arguments

Satisfying the condition for RVM arguments requires demonstrating the plausibility of being able to establish the three requisite properties of an RVM. The first property requires that no B-subject be allowed to access B-objects without those accesses being mediated by TCB subset B. The tamper resistance property requires that TCB subset A provide a way that TCB subset B can be designed and implemented such that A-subjects that are not part of B's implementation cannot tamper with B's policy-critical code or data. The third RVM property must be satisfied by the individual TCB subsets. The degree to which each TCB subset must satisfy this property is commensurate with the evaluation class of the TCB.

TC-6.4.2 EVALUATION CONSEQUENCES

In this example, the conditions for evaluation by parts are not satisfied and thus, the full potential for savings in evaluation effort cannot, in general, be realized. A clear option in such cases is to view the system as a monolithic TCB and proceed accordingly. However, because this case represents an example of an altered TCB, it admits of a wide spectrum of specific sub-cases. Thus, if the analysis of the system proceeds in parallel to that required for evaluation by parts it may be possible, in special cases, to identify elements of the analysis of the more primitive candidate TCB subset which can be successfully argued to be unaffected by the alterations. Some evaluation effort, often significant, can be saved, but unlike evaluation by parts, how much can only be estimated by consideration of the implementation specifics. For example, in this specific case, the local analysis of TCB subset C represents a reasonable candidate for analysis that need not be redone.

TC-6.5 SUMMARY

The three cases described above illustrate the effects of various TCB subsetting situations as they relate to the evaluation requirements.

A monolithic evaluation proceeds exactly as described in the TCSEC, with requirements being applied to the entire TCB.

When all the conditions for evaluation by parts are satisfied, considerable savings in evaluation effort are realized. The evaluation of a new system configuration that includes exactly the same TCB subset that was previously evaluated (such as the TCB subsets A and B in the Section TC-6.3) is limited to (a) local analysis of the individual TCB subsets (by reference to earlier analysis, if available) and (b) a simpler global analysis, because each TCB subset is an exact analog of a TCB.

When the conditions for evaluation by parts are not satisfied, no general statements can be made about the factorability of analysis or the reusability of previous analysis. The extent to which previous evaluation evidence and results remain valid can be determined only on a case-by-case basis.

PART 2

INTERPRETED REQUIREMENTS

IR-1 OVERVIEW AND CONTEXT

Part 2, "INTERPRETED REQUIREMENTS," provides specific interpretations of those TCSEC requirements which are deemed to be either DBMS-specific (or, more generally, application-specific) or particularly relevant to DBMSs. All of the requirements in the TCSEC apply in any case.

For the topics included below, the interpretations provide clarification of the TCSEC requirements. As is the case for the TCSEC, the interpreted requirements at any class include those specified for that class in addition to interpretations for lower classes that have not been superseded or altered.

Section IR-2 presents an overall summary of the TCSEC requirements, as interpreted in the more detailed sections that follow. Sections IR-3 through IR-7 address individual requirements interpretations for labels, audit, system architecture, design specification and verification, and design documentation, respectively. The format is an initial discussion of the topic in general, followed by specific requirements and interpretations that apply to database management systems. A listing of the requirements and interpretations organized by TCSEC class is presented in Appendix A.

IR-2 SUMMARY OF THE INTERPRETATIONS

This section provides specific interpretations of those TCSEC requirements that are particularly relevant for subsetted architectures and evaluation by parts. Its organization is derived from the structure of the TCSEC requirements. For each relevant TCSEC requirement there is a discussion of how that requirement is interpreted for a DBMS.

IR-2.1 SECURITY POLICY

IR-2.1.1 DISCRETIONARY ACCESS CONTROL

The requirement for discretionary access control is not changed in the context of this document and therefore applies as stated in the TCSEC.

IR-2.1.2 OBJECT REUSE

The requirement for object reuse is not changed in the context of this document and therefore applies as stated in the TCSEC.

IR-2.1.3 LABELS

The requirement for labels is treated in Section IR-3 of this document.

IR-2.1.4 MANDATORY ACCESS CONTROL

The requirement for mandatory access control is not changed in the context of this document and therefore applies as stated in the TCSEC. However, there are several subtle ramifications of this requirement of which a developer or evaluator should be aware. A brief discussion can be found in Appendix B, item 8, "Content-Dependent and Context-Dependent Access Control."

IR-2.2 ACCOUNTABILITY

IR-2.2.1 IDENTIFICATION AND AUTHENTICATION

The requirement for identification and authentication is not changed in the context of this document and therefore applies as stated in the TCSEC.

IR-2.2.2 AUDIT

The requirement for audit is treated in Section IR-4 of this document.

IR-2.3 ASSURANCE

IR-2.3.1 OPERATIONAL ASSURANCE

IR-2.3.1.1 System Architecture

The requirement for system architecture is treated in Section IR-5 of this document.

IR-2.3.1.2 System Integrity

The requirement for system integrity is not changed in the context of this document and therefore applies as stated in the TCSEC.

IR-2.3.1.3 Covert Channel Analysis

The requirement for covert channel analysis is not changed in the context of this document and therefore applies as stated in the TCSEC.

IR-2.3.1.4 Trusted Facility Management

The requirement for trusted facility management is not changed in the context of this document and therefore applies as stated in the TCSEC.

IR-2.3.1.5 Trusted Recovery

The requirement for trusted recovery is not changed in the context of this document and therefore applies as stated in the TCSEC.

IR-2.3.2 LIFE CYCLE ASSURANCE

IR-2.3.2.1 Security Testing

The requirement for security testing is not changed in the context of this document and therefore applies as stated in the TCSEC.

IR-2.3.2.2 Design Specification and Verification

The requirement for design specification and verification is treated in Section IR-6 of this document.

IR-2.3.2.3 Configuration Management

The requirement for configuration management is not changed in the context of this document and therefore applies as stated in the TCSEC.

IR-2.3.2.4 Trusted Distribution

The requirement for trusted distribution is not changed in the context of this document and therefore applies as stated in the TCSEC.

IR-2.4 DOCUMENTATION

IR-2.4.1 SECURITY FEATURES USER'S GUIDE

The requirement for a security features

user's guide is not changed in the context of this document and therefore applies as stated in the TCSEC.

IR-2.4.2 TRUSTED FACILITY MANUAL

The requirement for a trusted facility manual is not changed in the context of this document and therefore applies as stated in the TCSEC.

IR-2.4.3 TEST DOCUMENTATION

The requirement for test documentation is not changed in the context of this document and therefore applies as stated in the TCSEC.

IR-2.4.4 DESIGN DOCUMENTATION

The requirement for design documentation is treated in Section IR-7 of this document.

IR-3 LABELS

IR-3.1 GENERAL DISCUSSION

The labels requirements of the TCSEC are entirely applicable to database management systems. The basic difference between the TCSEC labeling requirements as they apply to operating systems and DBMSs involves which storage objects are labeled rather than how the labels are handled. This section discusses special considerations in implementing and evaluating labeling mechanisms in database management systems. Of particular importance is the selection of the storage objects that are to be labeled.

Beginning at the B1 evaluation class, trusted database management systems are required to associate labels with all storage objects under their control. The granularity of storage objects to be protected shall be chosen by the DBMS designer.

Stored view definitions (that is, named query commands) that are visible at the TCB boundary must be stored in labeled objects. The TCB must mediate access both to the view definition and to the view instantiation (that is, the set of labeled objects accessed as the result of executing the query command contained in the view definition).

It has been proposed in several designs that views be used as a mechanism to implement context- or content-dependent labeling. The intuitive attractiveness of this approach is undeniable, but the implementation details must be carefully worked out to achieve a sound implementation. A brief discussion of this topic can be found in Appendix B, item 8, "Content-Dependent and Context-Dependent Access Control."

TCB designers and evaluators may make distinctions between objects that are explicitly labeled or implicitly labeled. Such distinctions are meaningful only within the confines of the TCB; all storage objects are explicitly labeled from a point of view outside the TCB. For example, consider an object of one type (e.g., table or file) within the TCB that "contains" many (reference monitor) objects of another type (e.g., tuples and records). The file could have an explicit label associated with it, and some of the records could have explicit labels associated with them. Those records that have no explicit label would be implicitly labeled by the label of the file.

For database management systems, the objects that store the base data of the database (e.g., files, records, relations, and tuples), as well as those objects that store the metadata (e.g., directories, indices, schemas, data dictionaries, discretionary authorization tables, recovery logs, and transaction logs), must be labeled. Objects that need not be labeled include internal resources that are not user visible (e.g., printer daemon scratch files and resource allocation tables). The requirement for importing data (labeled and unlabeled) is the same as in the TCSEC. For additional information, see Appendix B, item 9, "Bulk Loading of a Database."

IR-3.2 SPECIFIC INTERPRETATIONS

CLASS (B1): LABELED SECURITY PROTECTION

There are no interpretations for this class.

CLASS (B2): STRUCTURED PROTECTION

Statement from TCSEC

Sensitivity labels associated with each ADP system resource . . . that is directly or indirectly accessible by subjects external to the TCB shall be maintained by the TCB.

Interpretation

Internal TCB variables that are not visible to untrusted subjects need not be labeled, provided that they are not directly or indirectly accessible by subjects external to the TCB. However, it is important to understand that such internal variables can function as covert signaling channels when untrusted subjects are able to detect changes in these variables by observing system behavior.

CLASS (B3): SECURITY DOMAINS

There are no additional requirements.

CLASS (A1): VERIFIED DESIGN

There are no additional requirements.

IR-4 AUDIT

IR-4.1 GENERAL DISCUSSION

The audit requirements of the TCSEC apply to database management systems. This section discusses special considerations in designing and evaluating audit mechanisms in database management systems.

The TCB must be capable of maintaining an audit trail of accesses and attempted accesses to the objects protected by the mandatory and discretionary security policies. Two examples are given to illustrate auditing techniques for discretionary access control decisions.

Example 1. Consider a DBMS TCB providing discretionary controls on defined views of the database. Because the named object presented at the TCB interface is the view definition (not the records instantiated through the view), all that needs to be auditable is the use of the view by any untrusted subject.

Example 2. Consider a DBMS TCB that enforces discretionary access control on individual data records. When a user enters a query, the construction of a response may involve a search over many records that are not returned to the user because they did not satisfy the query. Records that do satisfy the query but to which the user does not have access should be auditable. Records that do not satisfy the query need not be auditable. That is, in the context of audit, access permission to the user and satisfaction of a query are to be kept separate.

There is no need to audit operations that are strictly internal to the TCB. Separate security audit logs may be maintained by the operating system and the database management system. Likewise, separate identifications for the same user may be maintained by the operating system and the database management system. The correlation of separate audit logs may be done either at the time they are generated or at some later time.

The emphasis of the audit criterion is to provide individual accountability for actions by users. This goal is not the same as that for a backup and recovery log. There is no requirement to integrate the audit log with the backup and recovery log, although such an integrated log is not prohibited.

At the designer's discretion, there may be a selectable capability to reduce the number of audit records generated in response to queries that involve many access control decisions.

IR-4.2 SPECIFIC INTERPRETATIONS

CLASS (C2): CONTROLLED ACCESS PROTECTION

Statement from TCSEC

The TCB shall be able to create, maintain, and protect from modification . . . an audit trail of accesses to the objects it protects.

Interpretation

Auditable events, in the case of a database management system, are the individual operations initiated by untrusted users (e.g., updates, retrievals, and inserts), not just the invocation of the database management system. The auditing mechanism shall have the capability of auditing all mediated accesses which are visible to users. That is, each discretionary access control policy decision shall be auditable. Individual operations performed by trusted software, if totally transparent to the user, need not be auditable. If the total audit requirement is met by the use of more than one audit log, a method of correlation must be available.

CLASS (B1): LABELED SECURITY PROTECTION

Statement from TCSEC

The TCB shall be able to create, maintain, and protect from modification . . . an audit trail of accesses to the objects it protects.

Interpretation

Auditable events, in the case of a database management system, are the individual operations initiated by untrusted users (e.g., updates, retrievals, and inserts), not just the invocation of the database management system. The auditing mechanism shall have the capability of auditing all mediated accesses which are visible to the user. That is, each discretionary access control policy decision and each mandatory access control policy decision shall be auditable. Individual operations performed by trusted software, if totally transparent to the user, need not be auditable. If the total audit requirement is met by the use of more than one audit log, a method of correlation must be available.

CLASS (B2): STRUCTURED PROTECTION

There is no interpretation for the additional requirements.

CLASS (B3): SECURITY DOMAINS

There is no interpretation for the additional requirements.

CLASS (A1): VERIFIED DESIGN

There are no additional requirements.

IR-5 SYSTEM ARCHITECTURE

IR-5.1 GENERAL DISCUSSION

The system architecture requirements of the TCSEC apply to database management systems.

The interpretations provided are a duplication of the general interpreted requirements that apply to an evaluation by parts. They are included because DBMS evaluations often involve multiple TCB subsets.

IR-5.2 SPECIFIC INTERPRETATIONS

CLASS (C1): DISCRETIONARY SECURITY PROTECTION

Statement from TCSEC

The TCB shall maintain a domain for its own execution that protects it from external interference or tampering.

Interpretation

If the TCB is composed of multiple TCB subsets, this requirement applies to each TCB subset.

CLASS (C2): CONTROLLED ACCESS PROTECTION

There is no interpretation for the additional requirements.

CLASS (B1): LABELED SECURITY PROTECTION

There is no interpretation for the additional requirements.

CLASS (B2): STRUCTURED PROTECTION

Statement from TCSEC

The user interface to the TCB shall be completely defined and all elements of the TCB identified.

Interpretation

If the TCB is composed of multiple subsets, this requirement applies to the interface between the TCB subsets as well as the user interface to the whole TCB.

Statement from TCSEC

It shall make effective use of available hardware to separate those elements that are protection-critical from those that are not.

Interpretation

If the TCB is composed of multiple subsets, each TCB subset must make use of facilities provided to it by more primitive TCB subsets, such as support for execution domains and for distinct address spaces, to achieve the required separation.

CLASS (B3): SECURITY DOMAINS

There is no interpretation for the additional requirements.

CLASS (A1): VERIFIED DESIGN

There are no additional requirements.

IR-6 DESIGN SPECIFICATION AND VERIFICATION

IR-6.1 GENERAL DISCUSSION

The design specification and verification requirements of the TCSEC, with the related documentation requirements, apply to database management systems.

The interpretations provided include a duplication of general interpreted requirements that apply to an evaluation by parts. They are included because of the likelihood that a DBMS evaluation will involve multiple TCB subsets.

In the database context, the set of candidates for "subject" and "object" can be larger than normally encountered in trusted operating systems. Where a database system builds on an underlying trusted operating system, for example, the set of candidate subjects for the two TCB subsets includes both the active entities created by the operating system and those active entities created by the trusted portion of the DBMS. The set of candidates for objects is large. Examples are files, segments, buffers, structures, pages, relations, tables, tuples, rows, columns, elements, entities, relationships, procedures, metadata, system tables, query trees, query plans, locking mechanisms, rollback mechanisms, indices, recovery and backup mechanisms, precalculated operations (such as joins), view definitions, view instantiations, constraints, authorization tables, data dictionary tables, and audit logs.

The requirements in the TCSEC for showing how the various representations of the system being evaluated fit together can be represented as in Figure IR-1. For monolithic TCBs, the policy must be stated; the model must be developed, maintained, and shown to be sufficient to enforce the policy; and the DTLS (FTLS for A1) must be constructed and shown to correspond both to the model and to the TCB implementation. These steps allow a chain of reasoning to proceed from the stated policy to the assertion that the system in question actually enforces the policy.

In the case of multiple TCB subsets, the intent is the same. Tracing all policy requirements to the actual system implementation must be possible, and vice versa. The current dilemma is that the theory and techniques for subdividing a model into a set of models (or a top level specification into a set of top level specifications) have not yet been established. Likewise neither theory or techniques have been established for composing a set of models or top level specifications into a unified model or top level specification. The absence of rigorous methods does not preclude an evaluation using a subsetted TCB.

The process of mapping policy to implementation is possible for each TCB subset, using the same techniques required for a monolithic TCB. For subsetted TCBs, designers and evaluators must explicitly show how the policy, model and specifications for each TCB subset meet the TCSEC requirements. In addition, convincing informal arguments must be given to show how the collection of TCB subsets enforces the policy of the composite TCB. Because more rigorous composition methods are unavailable, convincing informal arguments are appropriate for evaluation of TCBs up to and including Class A1.

The TCSEC requirements concerning the mapping from policy to implementation for a TCB composed of multiple TCB subsets raise these crucial topics:

The allocation of policy to the TCB subsets,

The relation of the models for the TCB subsets to the overall system policy, and

The relation of the top level specification for each TCB subset to the entire system.

Allocation of policy to the TCB subsets is a precise division of the policy for the entire system, as addressed in the policy allocation condition of Section TC-4.3.

The second topic, above, requires that the policy for each TCB subset be stated. Additionally, it is required that there be an informal convincing argument that the collection of models represents the policy enforced by the entire system.

The third topic, the way in which the set of top level specifications for the individual TCB subsets describes the composite TCB interface with respect to exceptions, errors and effects, is treated in a similar fashion. The top level specifications for each TCB subset must meet the requirement. There is, in addition, a requirement for an informal, convincing description of how the set of top level specifications describes the TCB interface with respect to exceptions, errors, and effects. At the A1 level, there is no requirement for additional formal specification or formal proofs beyond the specification and proofs specific to the individual TCB subsets.

Rather than formally composing the policies, models, and specifications and performing a single monolithic evaluation, a series of separate evaluations may be performed (one for each TCB subset). The evaluations are then tied together by presentation of sufficient informal arguments that the individual policies collectively represent the policy enforced by the entire system, that the individual models collectively represent the system's policy, that the individual specifications represent the TCB interface, and that the source code of each TCB subset is consistent with its top level specification.

Note that satisfactory completion of these requirements is logically equivalent to demonstrating that a "unified" model for the entire TCB is consistent with the policy enforced by the system, that a "unified" top level specification corresponds to the model, and that the "unified" top level specification(s) corresponds to the source code. These interpreted requirements, which do not mandate a "unified" top level specification, are technically achievable interpretations of the policy-tracing requirements in the case of multiple TCB subsets.

IR-6.2 SPECIFIC INTERPRETATIONS

CLASS (B1): LABELED SECURITY PROTECTION

Statement from TCSEC

An informal or formal model of the security policy supported by the TCB shall be maintained over the life cycle of the ADP system and demonstrated to be consistent with its axioms.

Interpretation

If the TCB is composed of multiple TCB subsets, this requirement applies to the security policy of each TCB subset. An informal argument that the set of policy models represents the "security policy supported by the [composite] TCB" must be provided.

CLASS (B2): STRUCTURED PROTECTION

Statement from TCSEC

A formal model of the security policy supported by the TCB shall be maintained over the life cycle of the ADP system and demonstrated to be consistent with its axioms.

Interpretation

If the TCB is composed of multiple TCB subsets, this requirement applies to the security policy of each TCB subset. An informal argument that the set of policy models represents the "security policy supported by the [composite] TCB" must be provided.

Statement from TCSEC

A descriptive top-level specification (DTLS) of the TCB shall be maintained that completely and accurately describes the TCB in terms of exceptions, error messages, and effects.

Interpretation

If the TCB is composed of multiple TCB subsets, this requirement applies to the DTLS of each TCB subset. An informal argument that the set of DTLSs accurately describes the TCB must be provided.

If there is a multifaceted policy (e.g., both mandatory access control and discretionary access control policies) in a particular TCB subset, then all facets must be represented in the DTLS and in the TCB subset's model.

Statement from TCSEC

The descriptive top-level specification (DTLS) shall be shown to be an accurate description of the TCB interface.

Interpretation

If the TCB is composed of multiple subsets, this requirement applies to the interface between the TCB subsets as well as to the user interface of the composite TCB. The TCB interface description shall include an explanation of how to describe the total TCB accurately, in the context of the multiple TCB subset DTLSs.

CLASS (B3): SECURITY DOMAINS

Statement from TCSEC

A convincing argument shall be given that the DTLS is consistent with the model.

Interpretation

If the TCB is composed of multiple TCB subsets, this requirement applies to individual TCB subsets. Enforcement of all policies must be shown to occur in all situations (e.g., state transitions) required by the formal security policy model. In the case of a discretionary access control policy, the presence of the access control check at all identified state transitions is the total of what is required for demonstrating consistency between the DTLS and the model. This may be verified by inspection of the DTLS (that is, by visually checking that exception checks required by the model are present in the DTLS). For the mandatory access control policy, the DTLS must be shown by a convincing argument to be consistent with the model.

CLASS (A1): VERIFIED DESIGN

Statement from TCSEC

A formal top-level specification (FTLS) of the TCB shall be maintained that accurately describes the TCB in terms of exceptions, error messages, and effects.

Interpretation

If the TCB is composed of multiple TCB subsets, this requirement applies to the FTLS of each TCB subset. An informal argument that the set of FTLSs accurately describes the TCB must be provided.

If there is a multifaceted policy (e.g., both mandatory access control and discretionary access control policies) in a particular TCB subset, then all facets must be represented in the FTLS and in the TCB subset's model.

Statement from TCSEC

The FTLS shall be shown to be an accurate description of the TCB interface.

Interpretation

If the TCB is composed of multiple subsets, this requirement applies to the interface between the TCB subsets as well as to the user interface of the composite TCB. The TCB interface description shall include an explanation of how to describe the total TCB accurately, in the context of the multiple TCB subset DTLSs.

Statement from TCSEC

. . . a combination of formal and informal techniques shall be used to show that the FTLS is consistent with the model.

Interpretation

If the TCB is composed of multiple TCB subsets, this requirement applies to individual TCB subsets. Enforcement of all policies must be shown to occur in all situations (e.g., state transitions) required by the formal security policy model at the required occasions. In the case of a discretionary access control policy, the presence of the access control check at all identified state transitions is the total of what is required for demonstrating consistency between the FTLS and the model. This may be verified by inspection of the FTLS (that is, by visually checking that exception checks required by the model are present in the FTLS). For the mandatory access control policy, the FTLS must be shown by a combination of formal and informal techniques to be consistent with the model.

IR-7 DESIGN DOCUMENTATION

IR-7.1 GENERAL DISCUSSION

The design documentation requirement of the TCSEC applies to database management systems.

The interpretations provided are a duplication of the general interpreted requirements that apply to an evaluation by parts. They are included because DBMS evaluations often involve multiple TCB subsets.

IR-7.2 SPECIFIC INTERPRETATIONS

CLASS (C1): DISCRETIONARY SECURITY PROTECTION

Statement from TCSEC

If the TCB is composed of distinct modules, the interfaces between these modules shall be described.

Interpretation

If the TCB is composed of multiple subsets, this requirement applies to each TCB subset and the interfaces between TCB subsets.

CLASS (C2): CONTROLLED ACCESS PROTECTION

There are no additional requirements.

CLASS (B1): LABELED SECURITY PROTECTION

Statement from TCSEC

The specific TCB protection mechanisms shall be identified and an explanation given to show that they satisfy the model.

Interpretation

If the TCB is composed of multiple subsets, this requirement applies to each TCB subset and shall include the protection mechanisms which support the conditions for TCB subset structure and separate subset-domains.

CLASS (B2): STRUCTURED PROTECTION

Statement from TCSEC

The interfaces between the TCB modules shall be described.

Interpretation

If the TCB is composed of multiple subsets, this requirement applies to each TCB subset and the interfaces between different TCB subsets.

Statement from TCSEC

The descriptive top-level specification (DTLS) shall be shown to be an accurate description of the TCB interface.

Interpretation

If the TCB is composed of multiple subsets, this requirement applies to the interface between the TCB subsets as well as to the user interface of the composite TCB. The TCB interface description shall include an explanation of how to describe the total TCB accurately, in the context of the multiple TCB subset DTLSs.

Statement from TCSEC

Documentation shall describe how the TCB implements the reference monitor concept and give an explanation of why it is tamper resistant, cannot be bypassed, and is correctly implemented.

Interpretation

If the TCB is composed of multiple TCB subsets, this requirement is interpreted to apply to each TCB subset with respect to its specific technical policy. In addition, there must be documented an informal argument that the cooperative action of the TCB subsets makes the TCB itself tamper resistant, non-bypassable, and correct.

Statement from TCSEC

Documentation shall describe how the TCB is structured to facilitate testing and to enforce least privilege.

Interpretation

If the TCB is composed of multiple subsets, this requirement is interpreted to apply to individual TCB subsets as well as to the overall TCB.

CLASS (B3): SECURITY DOMAINS

Statement from TCSEC

The TCB implementation shall be informally shown to be consistent with the DTLS.

Interpretation

If the TCB is composed of multiple TCB subsets, this requirement is interpreted to apply to individual TCB subsets.

CLASS (A1): VERIFIED DESIGN

Statement from TCSEC

The TCB implementation shall be informally shown to be consistent with the FTLS.

Interpretation

If the TCB is composed of multiple TCB subsets, this requirement is interpreted to apply to individual TCB subsets.

APPENDIX A

SUMMARY OF THE
INTERPRETATIONS BY CLASS

This section is a presentation of all the interpreted requirements organized by TCSEC class. It includes all the requirements which are either relevant to subsetted architectures or are DBMS-specific. Any TCSEC requirements not explicitly identified herein apply as stated in the TCSEC.

CLASS (C1): DISCRETIONARY SECURITY PROTECTION

C1-1 SECURITY POLICY

C1-1.1 DISCRETIONARY ACCESS CONTROL

The discretionary access control requirements apply as stated in the TCSEC to every TCB subset whose policy includes discretionary access control of its subjects to its objects. Any TCB subset whose policy does not include such discretionary access control is exempt from this requirement.

C1-2 ACCOUNTABILITY

C1-2.1 IDENTIFICATION AND AUTHENTICATION

This requirement applies as stated in the TCSEC to the entire TCB. The cooperative action of the TCB subsets making up the TCB must satisfy the requirement.

If the TCB is composed of TCB subsets, one TCB subset may either rely on a mechanism in another TCB subset to provide identification and authentication services or provide the services directly. Each TCB subset may maintain its own identification and authentication data or one central repository may be maintained. If each TCB subset has its own data, then the information for each individual user must be consistent among all subsets.

C1-3 ASSURANCE

C1-3.1 OPERATIONAL ASSURANCE

C1-3.1.1 SYSTEM ARCHITECTURE

This requirement applies as stated in the

TCSEC to every TCB subset, with the following additional interpretations.

The TCB must provide domains for execution that are allocated to and used by TCB subsets according to the subset-domain condition for evaluation by parts. A most primitive TCB subset must provide domains for execution. A less primitive TCB subset must make use of domains provided by a more primitive TCB subset. A less primitive TCB subset may provide further execution domains but is not required to do so.

Statement from TCSEC

The TCB shall maintain a domain for its own execution that protects it from external interference or tampering.

Interpretation

If the TCB is composed of multiple TCB subsets, this requirement applies to each TCB subset.

C1-3.1.2 SYSTEM INTEGRITY

This requirement applies as stated in the TCSEC to every TCB subset that includes hardware or firmware. Any TCB subset that does not include hardware or firmware is exempt from this requirement.

C1-3.2 LIFE CYCLE ASSURANCE

C1-3.2.1 SECURITY TESTING

This requirement applies as stated in the TCSEC to the entire TCB. If a TCB consists of TCB subsets meeting the conditions for evaluation by parts, the satisfaction of the requirements by each TCB subset satisfies the requirement for the entire TCB. Otherwise, security testing of the entire TCB must be performed (even if the results of testing the individual TCB subsets were available).

C1-4 DOCUMENTATION

C1-4.1 SECURITY FEATURES USER'S GUIDE

This requirement applies as stated in the TCSEC to every TCB subset in the TCB. This collection of guides must include descriptions of every TCB subset in the TCB and explicit cross-references to other related user's guides to other TCB subsets, as required. In addition, interactions between mechanisms within different TCB subsets must be clearly described.

C1-4.2 TRUSTED FACILITY MANUAL

This requirement applies as stated in the TCSEC to the TCB and to every TCB subset in the TCB.

This requirement can be met by providing a set of manuals, one for each distinct (non-replicated) TCB subset. Each manual shall address the functions and privileges to be controlled for the associated TCB subset. Additionally, it must clearly show the interfaces to, and the interaction with, more primitive TCB subsets. The manual for each TCB subset shall identify the functions and privileges of the TCB subsets on which the associated TCB subset depends. Also, the TCB subset's manual shall identify which, if any, configuration options of the more primitive TCB subsets are to be used for the composite TCB to operate securely.

Any pre-defined roles supported (e.g., database administrator) by the TCB subset shall be addressed.

The means for correlating multiple audit logs and synonymous user identifications from multiple TCB subsets (if such exist) shall also be addressed.

The trusted facility manual shall describe the composite TCB so that the interactions among the TCB subsets can be readily determined. Other manuals may be referenced in this determination. The manuals that address the distinct modules of the TCB and the generation of the TCB need to be integrated with the other trusted facility manuals only to the extent that they are affected by the use and operation (versus the development) of the composite TCB.

C1-4.3 TEST DOCUMENTATION

This requirement applies as stated in the TCSEC to the composite TCB.

C1-4.4 DESIGN DOCUMENTATION

This requirement applies as stated in the TCSEC to the composite TCB.

Statement from TCSEC

If the TCB is composed of distinct modules, the interfaces between these modules shall be described.

Interpretation

If the TCB is composed of multiple subsets, this requirement applies to each TCB subset and the interfaces between TCB subsets.

CLASS (C2): CONTROLLED ACCESS PROTECTION

C2-1 SECURITY POLICY

C2-1.1 DISCRETIONARY ACCESS CONTROL

The discretionary access control requirements apply as stated in the TCSEC to every TCB subset whose policy includes discretionary access control of its subjects to its objects. Any TCB subset whose policy does not include such discretionary access control is exempt from this requirement.

C2-1.2 OBJECT REUSE

This requirement applies as stated in the TCSEC to every TCB subset in the TCB.

C2-2 ACCOUNTABILITY

C2-2.1 IDENTIFICATION AND AUTHENTICATION

This requirement applies as stated in the TCSEC to the entire TCB. The cooperative action of the TCB subsets making up the TCB must satisfy the requirement.

If the TCB is composed of TCB subsets, one TCB subset may either rely on a mechanism in another TCB subset to provide identification and authentication services or provide the services directly. Each TCB subset may maintain its own identification and authentication data or one central repository may be maintained. If each TCB subset has its own data, then the information for each individual user must be consistent among all subsets.

C2-2.2 AUDIT

This requirement applies as stated in the TCSEC to the entire TCB. The cooperative action of the TCB subsets making up the TCB must satisfy the requirement.

A TCB subset may maintain its own security audit log, distinct from that maintained by more primitive TCB subsets, or it may use an audit interface provided by a different TCB subset allowing the audit records it generates to be processed by that TCB subset.

If the TCB subset uses different user identifications than a more primitive TCB subset, there shall be a means to associate audit records generated by different TCB subsets for the same individual with each other, either at the time they are generated or later.

Statement from TCSEC

The TCB shall be able to create, maintain, and protect from modification . . . an audit trail of access to the objects it protects.

Interpretation

Auditable events, in the case of a database management system, are the individual operations initiated by untrusted users (e.g., updates, retrievals, and inserts), not just the invocation of the database management system. The auditing mechanism shall have the capability of auditing all mediated accesses which are visible to users. That is, each discretionary access control policy decision shall be auditable. Individual operations performed by trusted software, if totally transparent to the user, need not be auditable. If the total audit requirement is met by the use of more than one audit log, a method of correlation must be available.

C2-3 ASSURANCE

C2-3.1 OPERATIONAL ASSURANCE

C2-3.1.1 SYSTEM ARCHITECTURE

This requirement applies as stated in the TCSEC to every TCB subset, with the following additional interpretations.

The TCB must provide domains for execution that are allocated to and used by TCB subsets according to the subset-domain condition for evaluation by parts. A most primitive TCB subset must provide domains for execution. A less primitive TCB subset must make use of domains provided by a more primitive TCB subset. A less primitive TCB subset may provide further execution domains but is not required to do so.

Statement from TCSEC

The TCB shall maintain a domain for its own execution that protects it from external interference or tampering.

Interpretation

If the TCB is composed of multiple TCB subsets, this requirement applies to each TCB subset.

C2-3.1.2 SYSTEM INTEGRITY

This requirement applies as stated in the TCSEC to every TCB subset that includes hardware or firmware. Any TCB subset that does not include hardware or firmware is exempt from this requirement.

C2-3.2 LIFE CYCLE ASSURANCE

C2-3.2.1 SECURITY TESTING

This requirement applies as stated in the TCSEC to the entire TCB. If a TCB consists of TCB subsets meeting the conditions for evaluation by parts, the satisfaction of the requirements by each TCB subset satisfies the requirement for the entire TCB. Otherwise, security testing of the entire TCB must be performed (even if the results of testing the individual TCB subsets were available).

C2-4 DOCUMENTATION

C2-4.1 SECURITY FEATURES USER'S GUIDE

This requirement applies as stated in the TCSEC to every TCB subset in the TCB. This collection of guides must include descriptions of every TCB subset in the TCB and explicit cross-references to other related user's guides to other TCB subsets, as required. In addition, interactions between mechanisms within different TCB subsets must be clearly described.

C2-4.2 TRUSTED FACILITY MANUAL

This requirement applies as stated in the TCSEC to the TCB and to every TCB subset in the TCB.

This requirement can be met by providing a set of manuals, one for each distinct (non-replicated) TCB subset. Each manual shall address the functions and privileges to be controlled for the associated TCB subset. Additionally, it must clearly show the interfaces to, and the interaction with, more primitive TCB subsets. The manual for each TCB subset shall identify the functions and privileges of the TCB subsets on which the associated TCB subset depends. Also, the TCB subset's manual shall identify which, if any, configuration options of the more primitive TCB subsets are to be used for the composite TCB to operate securely.

Any pre-defined roles supported (e.g., database administrator) by the TCB subset shall be addressed.

The means for correlating multiple audit logs and synonymous user identifications from multiple TCB subsets (if such exist) shall also be addressed.

The trusted facility manual shall describe the composite TCB so that the interactions among the TCB subsets can be readily determined. Other manuals may be referenced in this determination. The manuals that address the distinct modules of the TCB and the generation of the TCB need to be integrated with the other trusted facility manuals only to the extent that they are affected by the use and operation (versus the development) of the composite TCB.

C2-4.3 TEST DOCUMENTATION

This requirement applies as stated in the TCSEC to the composite TCB.

C2-4.4 DESIGN DOCUMENTATION

This requirement applies as stated in the TCSEC to the composite TCB.

Statement from TCSEC

If the TCB is composed of distinct modules, the interface between these modules shall be described.

Interpretation

If the TCB is composed of multiple subsets, this requirement applies to each TCB subset and the interfaces between TCB subsets.

CLASS (B1): LABELED SECURITY PROTECTION

B1-1 SECURITY POLICY

B1-1.1 DISCRETIONARY ACCESS CONTROL

The discretionary access control requirements apply as stated in the TCSEC to every TCB subset whose policy includes discretionary access control of its subjects to its objects. Any TCB subset whose policy does not include such discretionary access control is exempt from this requirement.

B1-1.2 OBJECT REUSE

This requirement applies as stated in the TCSEC to every TCB subset in the TCB.

B1-1.3 LABELS

This requirement applies as stated in the TCSEC to every TCB subset whose policy includes mandatory access control of its subjects to its objects. Any TCB subset whose policy does not include such mandatory access control is exempt from this requirement.

B1-1.3.1 LABEL INTEGRITY

This requirement applies as stated in the TCSEC to every TCB subset whose policy includes mandatory access control of its subjects to its objects. Any TCB subset whose policy does not include such mandatory access control is exempt from this requirement.

B1-1.3.2 EXPORTATION OF LABELED INFORMATION

This requirement applies as stated in the TCSEC to every TCB subset whose policy includes mandatory access control of its subjects to its objects. Any TCB subset whose policy does not include such mandatory access control is exempt from this requirement.

B1-1.4 MANDATORY ACCESS CONTROL

This requirement applies as stated in the TCSEC to every TCB subset whose policy includes mandatory access control of its subjects to its objects. Any TCB subset whose policy does not include such mandatory access control is exempt from this requirement.

B1-2 ACCOUNTABILITY

B1-2.1 IDENTIFICATION AND AUTHENTICATION

This requirement applies as stated in the TCSEC to the entire TCB. The cooperative action of the TCB subsets making up the TCB must satisfy the requirement.

If the TCB is composed of TCB subsets, one TCB subset may either rely on a mechanism in another TCB subset to provide identification and authentication services or provide the services directly. Similarly, that TCB subset may rely on a mechanism in another more primitive TCB subset to ensure that the security level of subjects external to the TCB that may be created to act on behalf of the individual user are dominated by the clearance and authorization of that user. Each TCB subset may maintain its own identification and authentication data or one central repository may be maintained. If each TCB subset has its own data, then the information for each individual user must be consistent among all subsets.

B1-2.2 AUDIT

This requirement applies as stated in the TCSEC to the entire TCB. The cooperative action of the TCB subsets making up the TCB must satisfy the requirement.

A TCB subset may maintain its own security audit log, distinct from that maintained by more primitive TCB subsets, or it may use an audit interface provided by a different TCB subset allowing the audit records it generates to be processed by that TCB subset.

If the TCB subset uses different user identifications than a more primitive TCB subset, there shall be a means to associate audit records generated by different TCB subsets for the same individual with each other, either at the time they are generated or later.

Statement from TCSEC

The TCB shall be able to create, maintain, and protect from modification . . . an audit trail of access to the objects it protects.

Interpretation

Auditable events, in the case of a database management system, are the individual operations initiated by untrusted users (e.g., updates, retrievals, and inserts), not just the invocation of the database management system. The auditing mechanism shall have the capability of auditing all mediated accesses which are visible to users. That is, each discretionary access control policy decision shall be auditable. Individual operations performed by trusted software, if totally transparent to the user, need not be auditable. If the total audit requirement is met by the use of more than one audit log, a method of correlation must be available.

Statement from TCSEC

The TCB shall be able to create, maintain, and protect from modification . . . an audit trail of accesses to the objects it protects.

Interpretation

Auditable events, in the case of a database management system, are the individual operations initiated by untrusted users (e.g., updates, retrievals, and inserts), not just the invocation of the database management system. The auditing mechanism shall have the capability of auditing all mediated accesses which are visible to the user. That is, each discretionary access control policy decision and each mandatory access control policy decision shall be auditable. Individual operations performed by trusted software, if totally transparent to the user, need not be auditable. If the total audit requirement is met by the use of more than one audit log, a method of correlation must be available.

B1-3 ASSURANCE

B1-3.1 OPERATIONAL ASSURANCE

B1-3.1.1 SYSTEM ARCHITECTURE

This requirement applies as stated in the TCSEC to every TCB subset, with the following additional interpretations.

The TCB must provide domains for execution that are allocated to and used by TCB subsets according to the subset-domain condition for evaluation by parts. A most primitive TCB subset must provide domains for execution. A less primitive TCB subset must make use of domains provided by a more primitive TCB subset. A less primitive TCB subset may provide further execution domains but is not required to do so.

Similarly, the TCB must provide distinct address spaces for untrusted processes. A most primitive TCB subset must provide distinct address spaces for its subjects. A less primitive TCB subset must make use of the distinct address space provided by a more primitive TCB subset. A less primitive TCB subset may provide more fine-grained distinct address spaces, but is not required to do so.

Statement from TCSEC

The TCB shall maintain a domain for its own execution that protects it from external interference or tampering.

Interpretation

If the TCB is composed of multiple TCB subsets, this requirement applies to each TCB subset.

B1−3.1.2 SYSTEM INTEGRITY

This requirement applies as stated in the TCSEC to every TCB subset that includes hardware or firmware. Any TCB subset that does not include hardware or firmware is exempt from this requirement.

B1−3.1 LIFE CYCLE ASSURANCE

B1−3.2.1 SECURITY TESTING

This requirement applies as stated in the TCSEC to the entire TCB. If a TCB consists of TCB subsets meeting the conditions for evaluation by parts, the satisfaction of the requirements by each TCB subset satisfies the requirement for the entire TCB. Otherwise, security testing of the entire TCB must be performed (even if the results of testing the individual TCB subsets were available).

B1−3.2.2 DESIGN SPECIFICATION AND VERIFICATION

This requirement applies as stated in the TCSEC to every TCB subset, with the following specific additional interpretations.

It must be demonstrated that the security policy enforced by the composite TCB is represented by the collection of policy models for the individual TCB subsets.

Statement from TCSEC

An informal or formal model of the security policy supported by the TCB shall be maintained over the life cycle of the ADP system and demonstrated to be consistent with its axioms.

Interpretation

If the TCB is composed of multiple TCB subsets, this requirement applies to the security policy of each TCB subset. An informal argument that the set of policy models represents the "security policy supported by the [composite] TCB" must be provided.

B1-4 DOCUMENTATION

B1-4.1 SECURITY FEATURES USER'S GUIDE

This requirement applies as stated in the TCSEC to every TCB subset in the TCB. This collection of guides must include descriptions of every TCB subset in the TCB and explicit cross-references to other related user's guides to other TCB subsets, as required. In addition, interactions between mechanisms within different TCB subsets must be clearly described.

B1-4.2 TRUSTED FACILITY MANUAL

This requirement applies as stated in the TCSEC to the TCB and to every TCB subset in the TCB.

This requirement can be met by providing a set of manuals, one for each distinct (non-replicated) TCB subset. Each manual shall address the functions and privileges to be controlled for the associated TCB subset. Additionally, it must clearly show the interfaces to, and the interaction with, more primitive TCB subsets. The manual for each TCB subset shall identify the functions and privileges of the TCB subsets on which the associated TCB subset depends. Also, the TCB subset's manual shall identify which, if any, configuration options of the more primitive TCB subsets are to be used for the composite TCB to operate securely.

Any pre-defined roles supported (e.g., database administrator) by the TCB subset shall be addressed.

The means for correlating multiple audit logs and synonymous user identifications from multiple TCB subsets (if such exist) shall also be addressed.

The trusted facility manual shall describe the composite TCB so that the interactions among the TCB subsets can be readily determined. Other manuals may be referenced in this determination. The manuals that address the distinct modules of the TCB and the generation of the TCB need to be integrated with the other trusted facility manuals only to the extent that they are affected by the use and operation (versus the development) of the composite TCB.

B1-4.3 TEST DOCUMENTATION

This requirement applies as stated in the TCSEC to the composite TCB.

B1-4.4 DESIGN DOCUMENTATION

This requirement applies as stated in the TCSEC to the composite TCB, with the following specific additional interpretation:

Requirements concerning models and DTLSs apply to individual TCB subsets.

Statement from TCSEC

If the TCB is composed of distinct modules, the interface between these modules shall be described.

Interpretation

If the TCB is composed of multiple subsets, this requirement applies to each TCB subset and the interfaces between TCB subsets.

Statement from TCSEC

The specific TCB protection mechanisms shall be identified and an explanation given to show that they satisfy the model.

Interpretation

If the TCB is composed of multiple subsets, this requirement applies to each TCB subset and shall include the protection mechanisms which support the conditions for TCB subset structure and separate subset-domains.

CLASS (B2): STRUCTURED PROTECTION

B2-1 SECURITY POLICY

B2-1.1 DISCRETIONARY ACCESS CONTROL

The discretionary access control requirements apply as stated in the TCSEC to every TCB subset whose policy includes discretionary access control of its subjects to its objects. Any TCB subset whose policy does not include such discretionary access control is exempt from this requirement.

B2-1.2 OBJECT REUSE

This requirement applies as stated in the TCSEC to every TCB subset in the TCB.

B2-1.3 LABELS

This requirement applies as stated in the TCSEC to every TCB subset whose policy includes mandatory access control of its subjects to its objects. Any TCB subset whose policy does not include such mandatory access control is exempt from this requirement.

Statement from TCSEC

Sensitivity labels associated with each ADP system resource . . . that is directly or indirectly accessible by subjects external to the TCB shall be maintained by the TCB

Interpretation

Internal TCB variables that are not visible to untrusted subjects need not be labeled, provided that they are not directly or indirectly accessible by subjects external to the TCB. However, it is important to understand that such internal variables can function as covert signaling channels when untrusted subjects are able to detect changes in these variables by observing system behavior.

B2-1.3.1 LABEL INTEGRITY

This requirement applies as stated in the TCSEC to every TCB subset whose policy includes mandatory access control of its subjects to its objects. Any TCB subset whose policy does not include such mandatory access control is exempt from this requirement.

B2-1.3.2 EXPORTATION OF LABELED INFORMATION

This requirement applies as stated in the TCSEC to every TCB subset whose policy includes mandatory access control of its subjects to its objects. Any TCB subset whose policy does not include such mandatory access control is exempt from this requirement.

B2-1.3.3 SUBJECT SENSITIVITY LABELS

This requirement applies as stated in the TCSEC to the entire TCB. The cooperative action of the TCB subsets making up the TCB must satisfy the requirement.

B2-1.3.4 DEVICE LABELS

This requirement applies as stated in the TCSEC to every TCB subset whose policy includes mandatory access control of its subjects to its objects and has attached physical or virtual devices. Any TCB subset whose policy does not include such mandatory access control or has no attached physical or virtual devices is exempt from this requirement. This requirement can be satisifed by the cooperative action of more than one TCB subset.

B2-1.4 MANDATORY ACCESS CONTROL

This requirement applies as stated in the TCSEC to every TCB subset whose policy includes mandatory access control of its subjects to its objects. Any TCB subset whose policy does not include such mandatory access control is exempt from this requirement.

B2-2 ACCOUNTABILITY

B2-2.1 IDENTIFICATION AND AUTHENTICATION

This requirement applies as stated in the TCSEC to the entire TCB. The cooperative action of the TCB subsets making up the TCB must satisfy the requirement.

If the TCB is composed of TCB subsets, one TCB subset may either rely on a mechanism in another TCB subset to provide identification and authentication services or provide the services directly. Similarly, that TCB subset may rely on a mechanism in another more primitive TCB subset to ensure that the security level of subjects external to the TCB that may be created to act on behalf of the individual user are dominated by the clearance and authorization of that user. Each TCB subset may maintain its own identification and authentication data or one central repository may be maintained. If each TCB subset has its own data, then the information for each individual user must be consistent among all subsets.

B2-2.1.1 TRUSTED PATH

This requirement applies as stated in the TCSEC to the entire TCB. The cooperative action of the TCB subsets making up the TCB must satisfy the requirement.

When TCB subsets are used, the requirement for trusted path at levels B2 and above remains applicable to the entire TCB. The implementation of trusted path could be localized in a single TCB subset. Alternatively, it could be implemented in more than one TCB subset if the separate implementations together comply with the system security policy.

B2-2.2 AUDIT

This requirement applies as stated in the TCSEC to the entire TCB. The cooperative action of the TCB subsets making up the TCB must satisfy the requirement.

A TCB subset may maintain its own security audit log, distinct from that maintained by more primitive TCB subsets, or it may use an audit interface provided by a different TCB subset allowing the audit records it generates to be processed by that TCB subset.

If the TCB subset uses different user identifications than a more primitive TCB subset, there shall be a means to associate audit records generated by different TCB subsets for the same individual with each other, either at the time they are generated or later.

Any TCB subset wherein events may occur that require notification of the security administrator shall be able to: (1) detect the occurrence of these events, (2) initiate the recording of the audit trail entry, and (3) initiate the notification of the security administrator. The ability to terminate events (2) and (3) above may be provided either in the TCB subset within which they occur, or in the TCB subset(s) where actions that lead to the event were initiated.

The monitoring and notification requirements may require cooperation between multiple distinct TCB subsets or multiple instantiations of the same TCB subset. For example, to detect the accumulation of events for a single user it may be necessary to collect events from several subjects in distinct processes that are surrogates for the same user. As another example, there may be a single TCB subset that collects events from a number of other TCB subset instantiations and, based on its analysis of them, notifies the security administrator.

Statement from TCSEC

The TCB shall be able to create, maintain, and protect from modification . . . an audit trail of accesses to the objects it protects.

Interpretation

Auditable events, in the case of a database management system, are the individual operations initiated by untrusted users (e.g., updates, retrievals, inserts), not just the invocation of the database management system. The auditing mechanism shall have the capability of auditing all mediated accesses which are visible to the user. That is, each discretionary access control policy decision and each mandatory access control policy decision shall be auditable. Individual operations performed by trusted software, if totally transparent to the user, need not be auditable. If the total audit requirement is met by the use of more than one audit log, a method of correlation must be available.

Statement from TCSEC

The TCB shall be able to create, maintain, and protect from modification . . . an audit trail of accesses to the objects it protects.

Interpretation

Auditable events, in the case of a database management system, are the individual operations initiated by untrusted users (e.g., updates, retrievals, and inserts), not just the invocation of the database management system. The auditing mechanism shall have the capability of auditing all mediated accesses which are visible to the user. That is, each discretionary access control policy decision and each mandatory access control policy decision shall be auditable. Individual operations performed by trusted software, if totally transparent to the user, need not be auditable. If the total audit requirement is met by the use of more than one audit log, a method of correlation must be available.

B2-3 ASSURANCE

B2-3.1 OPERATIONAL ASSURANCE

B2-3.1.1 SYSTEM ARCHITECTURE

This requirement applies as stated in the TCSEC to every TCB subset, with the following additional interpretations.

The TCB must provide domains for execution that are allocated to and used by TCB subsets according to the subset-domain condition for evaluation by parts. A most primitive TCB subset must provide domains for execution. A less primitive TCB subset must make use of domains provided by a more primitive TCB subset. A less primitive TCB subset may provide further execution domains but is not required to do so.

Similarly, the TCB must provide distinct address spaces for untrusted processes. A most primitive TCB subset must provide distinct address spaces for its subjects. A less primitive TCB subset must make use of the distinct address space provided by a more primitive TCB subset. A less primitive TCB subset may provide more fine-grained distinct address spaces, but is not required to do so.

In general, requirements specifically referring to hardware or firmware apply only to TCB subsets that include hardware or firmware. The exception is the requirement that the TCB make effective use of available hardware. This requirement applies to those TCB subsets that use resources provided by more primitive TCB subsets in lieu of hardware. Those TCB subsets are required to make effective use of the protection-relevant features exported to it by the more primitive TCB subsets (e.g., execution domains, support for distinct address spaces) to separate those elements that are protection-critical from those that are not.

Statement from TCSEC

The TCB shall maintain a domain for its own execution that protects it from external interference or tampering.

Interpretation

If the TCB is composed of multiple TCB subsets, this requirement applies to each TCB subset.

Statement from TCSEC

The user interface to the TCB shall be completely defined and all elements of the TCB identified.

Interpretation

If the TCB is composed of multiple subsets, this requirement applies to the interface between the

TCB subsets as well as the user interface to the whole TCB.

Statement from TCSEC

It shall make effective use of available hardware to separate those elements that are protection-critical from those that are not.

Interpretation

If the TCB is composed of multiple subsets, each TCB subset must make use of facilities provided to it by more primitive TCB subsets, such as support for execution domains and for distinct address spaces, to achieve the required separation.

B2-3.1.2 SYSTEM INTEGRITY

This requirement applies as stated in the TCSEC to every TCB subset that includes hardware or firmware. Any TCB subset that does not include hardware or firmware is exempt from this requirement.

B2-3.1.3 COVERT CHANNEL ANALYSIS

This requirement applies as stated in the TCSEC to the entire TCB. When the TCB is made up entirely of TCB subsets meeting the conditions for evaluation by parts, analysis of the individual TCB subsets satisfies this requirement. Otherwise, covert channel analysis of the entire TCB must be performed (even if the results of covert channel analysis of the individual TCB subsets were available).

B2-3.1.4 TRUSTED FACILITY MANAGEMENT

This requirement applies as stated in the TCSEC to the entire TCB. Any "operator" or "administrator" functions intrinsic to an individual TCB subset must be supported by that TCB subset or by a more primitive TCB subset.

B2-3.2 LIFE CYCLE ASSURANCE

B2-3.2.1 SECURITY TESTING

This requirement applies as stated in the TCSEC to the entire TCB. If a TCB consists of TCB subsets meeting the conditions for evaluation by parts, the satisfaction of the requirements by each TCB subset satisfies the requirement for the entire TCB. Otherwise, security testing of the entire TCB must be performed (even if the results of testing the individual TCB subsets were available).

B2-3.2.2 DESIGN SPECIFICATION AND VERIFICATION

This requirement applies as stated in the TCSEC to every TCB subset, with the following specific additional interpretations.

It must be demonstrated that the security policy enforced by the composite TCB is represented by the collection of policy models for the individual TCB subsets.

The argument that the descriptive top level specification (DTLS) is consistent with the TCB interface applies to the entire TCB. There is required an explicit and convincing description of how the set of top level specifications (one for each TCB subset) describes the TCB interface in terms of exceptions, errors, and effects.

Statement from TCSEC

A formal model of the security policy supported by the TCB shall be maintained over the life cycle of the ADP system and demonstrated to be consistent with its axioms.

Interpretation

If the TCB is composed of multiple TCB subsets, this requirement applies to the security policy of each TCB subset. An informal argument that the set of policy models represents the "security policy supported by the [composite] TCB" must be provided.

Statement from TCSEC

A descriptive top-level specification (DTLS) of the TCB shall be maintained that completely and accurately describes the TCB in terms of exceptions, error messages, and effects.

Interpretation

If the TCB is composed of multiple TCB subsets, this requirement applies to the DTLS of each TCB subset. An informal argument that the set of DTLSs accurately describes the TCB must be provided.

If there is a multifaceted policy (e.g., both mandatory access control and discretionary access control policies) in a particular TCB subset, then all facets must be represented in the DTLS and in the TCB subset's model.

Statement from TCSEC

The descriptive top-level specification (DTLS) shall be shown to be an accurate description of the TCB interface.

Interpretation

If the TCB is composed of multiple subsets, this requirement applies to the interface between the TCB subsets as well as to the user interface of the composite TCB. The TCB interface description shall include an explanation of how to describe the total TCB accurately, in the context of the multiple TCB subset DTLSs.

B2-3.2.3 Configuration Management

This requirement applies as stated in the TCSEC to every TCB subset in the TCB, with the following additional interpretation.

Because subsets of the TCB may be developed independently, a single configuration management system may not be feasible. However, the combination of configuration management systems used to support all the TCB subsets must meet all the stated requirements. The information describing the interrelations between separate TCB subsets and separate security policy models falls into the category of "all documentation and code associated with the current version of the TCB" in the TCSEC requirements.

B2-4 DOCUMENTATION

B2-4.1 SECURITY FEATURES USER'S GUIDE

This requirement applies as stated in the TCSEC to every TCB subset in the TCB. This collection of guides must include descriptions of every TCB subset in the TCB and explicit cross-references to other related user's guides to other TCB subsets, as required. In addition, interactions between mechanisms within different TCB subsets must be clearly described.

B2-4.2 TRUSTED FACILITY MANUAL

This requirement applies as stated in the TCSEC to the TCB and to every TCB subset in the TCB.

This requirement can be met by providing a set of manuals, one for each distinct (non-replicated) TCB subset. Each manual shall address the functions and privileges to be controlled for the associated TCB subset. Additionally, it must clearly show the interfaces to, and the interaction with, more primitive TCB subsets. The manual for each TCB subset shall identify the functions and privileges of the TCB subsets on which the associated TCB subset depends. Also, the TCB subset's manual shall identify which, if any, configuration options of the more primitive TCB subsets are to be used for the composite TCB to operate securely.

Any pre-defined roles supported (e.g., database administrator) by the TCB subset shall be addressed.

The means for correlating multiple audit logs and synonymous user identifications from multiple TCB subsets (if such exist) shall also be addressed.

The trusted facility manual shall describe the composite TCB so that the interactions among the TCB subsets can be readily determined. Other manuals may be referenced in this determination. The manuals that address the distinct modules of the TCB and the generation of the TCB need to be integrated with the other trusted facility manuals only to the extent that they are affected by the use and operation (versus the development) of the composite TCB.

The TCB modules that contain the reference validation mechanism must be associated with the TCB subset to which they belong. The procedure for generating a new TCB after modifying only one (or several) TCB subsets must be described. This may be accommodated by independent regeneration of the individual TCB subsets or by regeneration of only the affected TCB subsets.

B2-4.3 TEST DOCUMENTATION

This requirement applies as stated in the TCSEC to the composite TCB.

B2-4.4 DESIGN DOCUMENTATION

This requirement applies as stated in the TCSEC to the composite TCB, with the following specific addditional interpretations.

Requirements concerning models and DTLSs apply to individual TCB subsets.

The requirement concerning the description of interfaces between modules of the TCB includes the interfaces between TCB subsets.

The documentation of the implementation of a reference validation mechanism must include justification for meeting the conditions for evaluation by parts.

Statement from TCSEC

The interfaces between the TCB modules shall be described.

Interpretation

If the TCB is composed of multiple subsets, this requirement applies to each TCB subset and the interfaces between TCB subsets.

Statement from TCSEC

The specific TCB protection mechanisms shall be identified and an explanation given to show that they satisfy the model.

Interpretation

If the TCB is composed of multiple subsets, this requirement applies to each TCB subset and shall include the protection mechanisms which support the conditions for TCB subset structure and separate subset-domains.

Statement from TCSEC

The descriptive top-level specification (DTLS) shall be shown to be an accurate description of the TCB interface.

Interpretation

If the TCB is composed of multiple subsets, this requirement applies to the interface between the TCB subsets as well as to the user interface of the composite TCB. The TCB interface description shall include an explanation of how to describe the total TCB accurately, in the context of the multiple TCB subset DTLSs.

Statement from TCSEC

Documentation shall describe how the TCB implements the reference monitor concept and give an explanation of why it is tamper resistant, cannot be bypassed, and is correctly implemented.

Interpretation

If the TCB is composed of multiple TCB subsets, this requirement is interpreted to apply to each TCB subset with respect to its specific technical policy. In addition, there must be documented an informal argument that the cooperative action of the TCB subsets makes the TCB itself tamper resistant, non-bypassable, and correct.

Statement from TCSEC

Documentation shall describe how the TCB is structured to facilitate testing and to enforce least privilege.

Interpretation

If the TCB is composed of multiple subsets, this requirement is interpreted to apply to individual TCB subsets as well as to the overall TCB.

CLASS (B3): SECURITY DOMAINS

B3-1 SECURITY POLICY

B3-1.1 DISCRETIONARY ACCESS CONTROL

The discretionary access control requirements apply as stated in the TCSEC to every TCB subset whose policy includes discretionary access control of its subjects to its objects. Any TCB subset whose policy does not include such discretionary access control is exempt from this requirement.

B3-1.2 OBJECT REUSE

This requirement applies as stated in the TCSEC to every TCB subset in the TCB.

B3-1.3 LABELS

This requirement applies as stated in the TCSEC to every TCB subset whose policy includes mandatory access control of its subjects to its objects. Any TCB subset whose policy does not include such mandatory access control is exempt from this requirement.

Statement from TCSEC

Sensitivity labels associated with each ADP system resource . . . that is directly or indirectly accessible by subjects external to the TCB shall be maintained by the TCB

Interpretation

Internal TCB variables that are not visible to untrusted subjects need not be labeled, provided that they are not directly or indirectly accessible by subjects external to the TCB. However, it is important to understand that such internal variables can function as covert signaling channels when untrusted subjects are able to detect changes in these variables by observing system behavior.

B3-1.3.1 LABEL INTEGRITY

This requirement applies as stated in the TCSEC to every TCB subset whose policy includes mandatory access control of its subjects to its objects. Any TCB subset whose policy does not include such mandatory access control is exempt from this requirement.

B3-1.3.2 EXPORTATION OF LABELED INFORMATION

This requirement applies as stated in the TCSEC to every TCB subset whose policy includes mandatory access control of its subjects to its objects. Any TCB subset whose policy does not include such mandatory access control is exempt from this requirement.

B3-1.3.3 SUBJECT SENSITIVITY LABELS

This requirement applies as stated in the TCSEC to the entire TCB. The cooperative action of the TCB subsets making up the TCB must satisfy the requirement.

B3-1.3.4 DEVICE LABELS

This requirement applies as stated in the TCSEC to every TCB subset whose policy includes mandatory access control of its subjects to its objects and has attached physical or virtual devices. Any TCB subset whose policy does not include such mandatory access control or has no attached physical or virtual devices is exempt from this requirement. This requirement can be satisifed by the cooperative action of more than one TCB subset.

B3-1.4 MANDATORY ACCESS CONTROL

This requirement applies as stated in the TCSEC to every TCB subset whose policy includes mandatory access control of its subjects to its objects. Any TCB subset whose policy does not include such mandatory access control is exempt from this requirement.

B3-2 ACCOUNTABILITY

B3-2.1 IDENTIFICATION AND AUTHENTICATION

This requirement applies as stated in the TCSEC to the entire TCB. The cooperative action of the TCB subsets making up the TCB must satisfy the requirement.

If the TCB is composed of TCB subsets, one TCB subset may either rely on a mechanism in another TCB subset to provide identification and authentication services or provide the services directly. Similarly, that TCB subset may rely on a mechanism in another more primitive TCB subset to ensure that the security level of subjects external to the TCB that may be created to act on behalf of the individual user are dominated by the clearance and authorization of that user. Each TCB subset may maintain its own identification and authentication data or one central repository may be maintained. If each TCB subset has its own data, then the information for each individual user must be consistent among all subsets.

B3-2.1.1 TRUSTED PATH

This requirement applies as stated in the TCSEC to the entire TCB. The cooperative action of the TCB subsets making up the TCB must satisfy the requirement.

When TCB subsets are used, the requirement for trusted path at levels B2 and above remains applicable to the entire TCB. The need for trusted path "when positive TCB-to-user connection is required (e.g., login, change subject security level)" can require user interaction with virtually any TCB subset within the TCB. The implementation of trusted path could be localized in a single TCB subset. Alternatively, it could be implemented in more than one TCB subset if the separate implementations together comply with the system security policy.

B3-2.2 AUDIT

This requirement applies as stated in the TCSEC to the entire TCB. The cooperative action of the TCB subsets making up the TCB must satisfy the requirement.

A TCB subset may maintain its own security audit log, distinct from that maintained by more primitive TCB subsets, or it may use an audit interface provided by a different TCB subset allowing the audit records it generates to be processed by that TCB subset.

If the TCB subset uses different user identifications than a more primitive TCB subset, there shall be a means to associate audit records generated by different TCB subsets for the same individual with each other, either at the time they are generated or at some later time.

Any TCB subset wherein events may occur that require notification of the security administrator shall be able to: (1) detect the occurrence of these events, (2) initiate the recording of the audit trail entry, and (3) initiate the notification of the security administrator. The ability to terminate events (2) and (3) above may be provided either in the TCB subset within which they occur, or in the TCB subset(s) where actions that lead to the event were initiated.

The monitoring and notification requirements may require cooperation between multiple distinct TCB subsets or multiple instantiations of the same TCB subset. For example, to detect the accumulation of events for a single user it may be necessary to collect events from several subjects in distinct processes that are surrogates for the same user. As another example, there may be a single TCB subset that collects events from a number of other TCB subset instantiations and, based on its analysis of them, notifies the security administrator.

Statement from TCSEC

The TCB shall be able to create, maintain, and protect from modification . . . an audit trail of accesses to the objects it protects.

Interpretation

Auditable events, in the case of a database management system, are the individual operations initiated by untrusted users (e.g., updates, retrievals, inserts), not just the invocation of the database management system. The auditing mechanism shall have the capability of auditing all mediated accesses which are visible to the user. That is, each discretionary access control policy decision and each mandatory access control policy decision shall be auditable. Individual operations performed by trusted software, if totally transparent to the user, need not be audited. If the total audit requirement is met by the use of more than one audit log, a method of correlation must be available.

Statement from TCSEC

The TCB shall be able to create, maintain, and protect from modification . . . an audit trail of accesses to the objects it protects.

Interpretation

Auditable events, in the case of a database management system, are the individual operations initiated by untrusted users (e.g., updates, retrievals, and inserts), not just the invocation of the database management system. The auditing mechanism shall have the capability of auditing all mediated accesses which are visible to the user. That is, each discretionary access control policy decision and each mandatory access control policy decision shall be auditable. Individual operations performed by trusted software, if totally transparent to the user, need not be auditable. If the total audit requirement is met by the use of more than one audit log, a method of correlation must be available.

B3-3 ASSURANCE

B3-3.1 OPERATIONAL ASSURANCE

B3-3.1.1 System Architecture

This requirement applies as stated in the TCSEC to every TCB subset, with the following additional interpretations.

The TCB must provide domains for execution that are allocated to and used by TCB subsets according to the subset-domain condition for evaluation by parts. A most primitive TCB subset must provide domains for execution. A less primitive TCB subset must make use of domains provided by a more primitive TCB subset. A less primitive TCB subset may provide further execution domains but is not required to do so.

Similarly, the TCB must provide distinct address spaces for untrusted processes. A most primitive TCB subset must provide distinct address spaces for its subjects. A less primitive TCB subset must make use of the distinct address space provided by a more primitive TCB subset. A less primitive TCB subset may provide more fine-grained distinct address spaces, but is not required to do so.

In general, requirements specifically referring to hardware or firmware apply only to TCB subsets that include hardware or firmware. However, the requirement that the TCB make effective use of hardware requires that a less primitive TCB subset make effective use of the protection-relevant features exported to it by the more primitive TCB subsets (e.g., execution domains, support for distinct address spaces) to separate those elements that are protection-critical from those that are not.

Statement from TCSEC

The TCB shall maintain a domain for its own execution that protects it from external interference or tampering.

Interpretation

If the TCB is composed of multiple TCB subsets, this requirement applies to each TCB subset.

Statement from TCSEC

The user interface to the TCB shall be completely defined and all elements of the TCB identified.

Interpretation

If the TCB is composed of multiple subsets, this requirement applies to the interface between the TCB subsets as well as the user interface to the whole TCB.

Statement from TCSEC

It shall make effective use of available hardware to separate those elements that are protection-critical from those that are not.

Interpretation

If the TCB is composed of multiple subsets, each TCB subset must make use of facilities provided to it by more primitive TCB subsets, such as support for execution domains and for distinct address spaces, to achieve the required separation.

B3-3.1.2 SYSTEM INTEGRITY

This requirement applies as stated in the TCSEC to every TCB subset that includes hardware or firmware. Any TCB subset that does not include hardware or firmware is exempt from this requirement.

B3-3.1.3 COVERT CHANNEL ANALYSIS

This requirement applies as stated in the TCSEC to the entire TCB. When the TCB is made up entirely of TCB subsets meeting the conditions for evaluation by parts, analysis of the individual TCB subsets satisfies this requirement. Otherwise, covert channel analysis of the entire TCB must be performed (even if the results of covert channel analysis of the individual TCB subsets were available).

B3-3.1.4 TRUSTED FACILITY MANAGEMENT

This requirement applies as stated in the TCSEC to the entire TCB. Any "operator" or "administrator" functions intrinsic to an individual TCB subset must be supported by that TCB subset or by a more primitive TCB subset.

B3-3.1.5 TRUSTED RECOVERY

This requirement applies as stated in the TCSEC to the entire TCB and to the individual TCB subsets. The cooperative recovery actions of the TCB subsets making up the TCB must provide trusted recovery for the overall TCB. Otherwise, trusted recovery evaluation must address the entire TCB (even if the individual TCB subsets meet the trusted recovery requirements).

B3-3.2 LIFE CYCLE ASSURANCE

B3-3.2.1 SECURITY TESTING

This requirement applies as stated in the TCSEC to the entire TCB. If a TCB consists of TCB subsets meeting the conditions for evaluation by parts, the satisfaction of the requirements by each TCB subset satisfies the requirement for the entire TCB. Otherwise, security testing of the entire TCB must be performed (even if the results of testing the individual TCB subsets were available).

B3-3.2.2 DESIGN SPECIFICATION AND VERIFICATION

This requirement applies as stated in the TCSEC to every TCB subset, with the following specific additional interpretations.

It must be demonstrated that the security policy enforced by the composite TCB is represented by the collection of policy models for the individual TCB subsets.

The argument that the descriptive top level specification (DTLS) is consistent with the TCB interface applies to the entire TCB. There is required an explicit and convincing description of how the set of top level specifications (one for each TCB subset) describes the TCB interface in terms of exceptions, errors, and effects.

Statement from TCSEC

A formal model of the security policy supported by the TCB shall be maintained over the life cycle of the ADP system and demonstrated to be consistent with its axioms.

Interpretation

If the TCB is composed of multiple TCB subsets, this requirement applies to the security policy of each TCB subset. An informal argument that the set of policy models represents the "security policy supported by the [composite] TCB" must be provided.

Statement from TCSEC

A descriptive top-level specification (DTLS) of the TCB shall be maintained that completely and accurately describes the TCB in terms of exceptions, error messages, and effects.

Interpretation

If the TCB is composed of multiple TCB subsets, this requirement applies to the DTLS of each TCB subset. An informal argument that the set of DTLSs accurately describes the TCB must be provided.

If there is a multifaceted policy (e.g., both mandatory access control and discretionary access control policies) in a particular TCB subset, then all facets must be represented in the DTLS and in the TCB subset's model.

Statement from TCSEC

The descriptive top-level specification (DTLS) shall be shown to be an accurate description of the TCB interface.

Interpretation

If the TCB is composed of multiple subsets, this requirement applies to the interface between the TCB subsets as well as to the user interface of the composite TCB. The TCB interface description shall include an explanation of how to describe the total TCB accurately, in the context of the multiple TCB subset DTLSs.

Statement from TCSEC

A convincing argument shall be given that the DTLS is consistent with the model.

Interpretation

If the TCB is composed of multiple TCB subsets, this requirement applies to individual TCB subsets. Enforcement of all policies must be shown to occur in all situations (e.g., state transitions) required by the formal security policy model. In the case of a discretionary access control policy, the presence of the access control check at all identified state transitions is the total of what is required for demonstrating consistency between the DTLS and the model. This may be verified by inspection of the DTLS (that is, by visually checking that exception checks required by the model are present in the DTLS). For the mandatory access control policy, the DTLS must be shown by a convincing argument to be consistent with the model.

B3-3.2.3 CONFIGURATION MANAGEMENT

This requirement applies as stated in the TCSEC to every TCB subset in the TCB, with the following additional interpretation.

Because subsets of the TCB may be developed independently, a single configuration management system may not be feasible. However, the combination of configuration management systems used to support all the TCB subsets must meet all the stated requirements. The information describing the interrelations between separate TCB subsets and separate security policy models falls into the category of "all documentation and code associated with the current version of the TCB" in the TCSEC requirements.

B3-4 DOCUMENTATION

B3-4.1 SECURITY FEATURES USER'S GUIDE

This requirement applies as stated in the TCSEC to every TCB subset in the TCB. This collection of guides must include descriptions of every TCB subset in the TCB and explicit cross-references to other related user's guides to other TCB subsets, as required. In addition, interactions between mechanisms within different TCB subsets must be clearly described.

B3-4.2 TRUSTED FACILITY MANUAL

This requirement applies as stated in the TCSEC to the TCB and to every TCB subset in the TCB.

This requirement can be met by providing a set of manuals, one for each distinct (non-replicated) TCB subset. Each manual shall address the functions and privileges to be controlled for the associated TCB subset. Additionally, it must clearly show the interfaces to, and the interaction with, more primitive TCB subsets. The manual for each TCB subset shall identify the functions and privileges of the TCB subsets on which the associated TCB subset depends. Also, the TCB subset's manual shall identify which, if any, configuration options of the more primitive TCB subsets are to be used for the composite TCB to operate securely.

Any pre-defined roles supported (e.g., database administrator) by the TCB subset shall be addressed.

The means for correlating multiple audit logs and synonymous user identifications from multiple TCB subsets (if such exist) shall also be addressed.

The trusted facility manual shall describe the composite TCB so that the interactions among the TCB subsets can be readily determined. Other manuals may be referenced in this determination. The manuals that address the distinct modules of the TCB and the generation of the TCB need to be integrated with the other trusted facility manuals only to the extent that they are affected by the use and operation (versus the development) of the composite TCB.

The TCB modules that contain the reference validation mechanism must be associated with the TCB subset to which they belong. The procedure for generating a new TCB after modifying only one (or several) TCB subsets must be described. This may be accommodated by independent regeneration of the individual TCB subsets or by regeneration of only the affected TCB subsets.

B3-4.3 TEST DOCUMENTATION

This requirement applies as stated in the TCSEC to the composite TCB.

B3-4.4 DESIGN DOCUMENTATION

This requirement applies as stated in the TCSEC to the composite TCB, with the following addtional specific interpretations.

Requirements concerning models and DTLSs apply to individual TCB subsets.

The requirement concerning the description of interfaces between modules of the TCB includes the interfaces between TCB subsets.

The documentation of the implementation of a reference validation mechanism must include justification for meeting the conditions for evaluation by parts.

Statement from TCSEC

The interfaces between the TCB modules shall be described.

Interpretation

If the TCB is composed of multiple subsets, this requirement applies to each TCB subset and the interfaces between TCB subsets.

Statement from TCSEC

The specific TCB protection mechanisms shall be identified and an explanation given to show that they satisfy the model.

Interpretation

If the TCB is composed of multiple subsets, this requirement applies to each TCB subset and shall include the protection mechanisms which support the conditions for TCB subset structure and separate subset-domains.

Statement from TCSEC

The descriptive top-level specification (DTLS) shall be shown to be an accurate description of the TCB interface.

Interpretation

If the TCB is composed of multiple subsets, this requirement applies to the interface between the TCB subsets as well as to the user interface of the composite TCB. The TCB interface description shall include an explanation of how to describe the total TCB accurately, in the context of the multiple TCB subset DTLSs.

Statement from TCSEC

Documentation shall describe how the TCB implements the reference monitor concept and give an explanation of why it is tamper resistant, cannot be bypassed, and is correctly implemented.

Interpretation

If the TCB is composed of multiple TCB subsets, this requirement is interpreted to apply to each TCB subset with respect to its specific technical policy. In addition, there must be documented an informal argument that the cooperative action of the TCB subsets makes the TCB itself tamper resistant, non-bypassable, and correct.

Statement from TCSEC

Documentation shall describe how the TCB is structured to facilitate testing and to enforce least privilege.

Interpretation

If the TCB is composed of multiple subsets, this requirement is interpreted to apply to individual TCB subsets as well as to the overall TCB.

Statement from TCSEC

The TCB implementation shall be informally shown to be consistent with the DTLS.

Interpretation

If the TCB is composed of multiple TCB

subsets, this requirement is interpreted to apply to individual TCB subsets.

CLASS (A1): VERIFIED DESIGN

A1-1 SECURITY POLICY

A1-1.1 DISCRETIONARY ACCESS CONTROL

The discretionary access control requirements apply as stated in the TCSEC to every TCB subset whose policy includes discretionary access control of its subjects to its objects. Any TCB subset whose policy does not include such discretionary access control is exempt from this requirement.

A1-1.2 OBJECT REUSE

This requirement applies as stated in the TCSEC to every TCB subset in the TCB.

A1-1.3 LABELS

This requirement applies as stated in the TCSEC to every TCB subset whose policy includes mandatory access control of its subjects to its objects. Any TCB subset whose policy does not include such mandatory access control is exempt from this requirement.

Statement from TCSEC

Sensitivity labels associated with each ADP system resource . . . that is directly or indirectly accessible by subjects external to the TCB shall be maintained by the TCB

Interpretation

Internal TCB variables that are not visible to untrusted subjects need not be labeled, provided that they are not directly or indirectly accessible by subjects external to the TCB. However, it is important to understand that such internal variables can function as covert signaling channels when untrusted subjects are able to detect changes in these variables by observing system behavior.

A1-1.3.1 LABEL INTEGRITY

This requirement applies as stated in the TCSEC to every TCB subset whose policy includes mandatory access control of its subjects to its objects. Any TCB subset whose policy does not include such mandatory access control is exempt from this requirement.

A1-1.3.2 EXPORTATION OF LABELED INFORMATION

This requirement applies as stated in the TCSEC to every TCB subset whose policy includes mandatory access control of its subjects to its objects. Any TCB subset whose policy does not include such mandatory access control is exempt from this requirement.

A1-1.3.3 SUBJECT SENSITIVITY LABELS

This requirement applies as stated in the TCSEC to the entire TCB. The cooperative action of the TCB subsets making up the TCB must satisfy the requirement.

A1-1.3.4 DEVICE LABELS

This requirement applies as stated in the TCSEC to every TCB subset whose policy includes mandatory access control of its subjects to its objects and has attached physical or virtual devices. Any TCB subset whose policy does not include such mandatory access control or has no attached physical or virtual devices is exempt from this requirement. This requirement can be satisifed by the cooperative action of more than one TCB subset.

A1-1.4 MANDATORY ACCESS CONTROL

This requirement applies as stated in the TCSEC to every TCB subset whose policy includes mandatory access control of its subjects to its objects. Any TCB subset whose policy does not include such mandatory access control is exempt from this requirement.

A1-2 ACCOUNTABILITY

A1-2.1 IDENTIFICATION AND AUTHENTICATION

This requirement applies as stated in the TCSEC to the entire TCB. The cooperative action of the TCB subsets making up the TCB must satisfy the requirement.

If the TCB is composed of TCB subsets, one TCB subset may either rely on a mechanism in another TCB subset to provide identification and authentication services or provide the services directly. Similarly, that TCB subset may rely on a mechanism in another more primitive TCB subset to ensure that the security level of subjects external to the TCB that may be created to act on behalf of the individual user are dominated by the clearance and authorization of that user. Each TCB subset may maintain its own identification and authentication data or one central repository may be maintained. If each TCB subset has its own data, then the information for each individual user must be consistent among all subsets.

A1-2.1.1 TRUSTED PATH

This requirement applies as stated in the TCSEC to the entire TCB. The cooperative action of the TCB subsets making up the TCB must satisfy the requirement.

When TCB subsets are used, the requirement for trusted path at levels B2 and above remains applicable to the entire TCB. The need for trusted path "when positive TCB-to-user connection is required (e.g., login, change subject security level)" can require user interaction with virtually any TCB subset within the TCB. The implementation of trusted path could be localized in a single TCB subset. Alternatively, it could be implemented in more than one TCB subset if the separate implementations together comply with the system security policy.

A1-2.2 AUDIT

This requirement applies as stated in the TCSEC to the entire TCB. The cooperative action of the TCB subsets making up the TCB must satisfy the requirement.

A TCB subset may maintain its own security audit log, distinct from that maintained by more primitive TCB subsets, or it may use an audit interface provided by a different TCB subset allowing the audit records it generates to be processed by that TCB subset.

If the TCB subset uses different user identifications than a more primitive TCB subset, there shall be a means to associate audit records generated by different TCB subsets for the same individual with each other, either at the time they are generated or at some later time.

Any TCB subset wherein events may occur that require notification of the security administrator shall be able to: (1) detect the occurrence of these events, (2) initiate the recording of the audit trail entry, and (3) initiate the notification of the security administrator. The ability to terminate events (2) and (3) above may be provided either in the TCB subset within which they occur, or in the TCB subset(s) where actions that lead to the event were initiated.

The monitoring and notification requirements may require cooperation between multiple distinct TCB subsets or multiple instantiations of the same TCB subset. For example, to detect the accumulation of events for a single user it may be necessary to collect events from several subjects in distinct processes that are surrogates for the same user. As another example, there may be a single TCB subset that collects events from a number of other TCB subset instantiations and, based on its analysis of them, notifies the security administrator.

Statement from TCSEC

The TCB shall be able to create, maintain, and protect from modification . . . an audit trail of accesses to the objects it protects.

Interpretation

Auditable events, in the case of a database management system, are the individual operations initiated by untrusted users (e.g., updates, retrievals, inserts), not just the invocation of the database management system. The auditing mechanism shall have the capability of auditing all mediated accesses which are visible to the user. That is, each discretionary access control policy decision and each mandatory access control policy decision shall be auditable. Individual operations performed by trusted software, if totally transparent to the user, need not be audited. If the total audit requirement is met by the use of more than one audit log, a method of correlation must be available.

Statement from TCSEC

The TCB shall be able to create, maintain, and protect from modification . . . an audit trail of accesses to the objects it protects.

Interpretation

Auditable events, in the case of a database management system, are the individual operations initiated by untrusted users (e.g., updates, retrievals, and inserts), not just the invocation of the database management system. The auditing mechanism shall have the capability of auditing all mediated accesses which are visible to the user. That is, each discretionary access control policy decision and each mandatory access control policy decision shall be auditable. Individual operations performed by trusted software, if totally transparent to the user, need not be auditable. If the total audit requirement is met by the use of more than one audit log, a method of correlation must be available.

A1-3 ASSURANCE

A1-3.1 OPERATIONAL ASSURANCE

A1-3.1.1 SYSTEM ARCHITECTURE

This requirement applies as stated in the TCSEC to every TCB subset, with the following additional interpretations.

The TCB must provide domains for execution that are allocated to and used by TCB subsets according to the subset-domain condition for evaluation by parts. A most primitive TCB subset must provide domains for execution. A less primitive TCB subset must make use of domains provided by a more primitive TCB subset. A less primitive TCB subset may provide further execution domains but is not required to do so.

Similarly, the TCB must provide distinct address spaces for untrusted processes. A most primitive TCB subset must provide distinct address spaces for its subjects. A less primitive TCB subset must make use of the distinct address space provided by a more primitive TCB subset. A less primitive TCB subset may provide more fine-grained distinct address spaces, but is not required to do so.

In general, requirements specifically referring to hardware or firmware apply only to TCB subsets that include hardware or firmware. However, the requirement that the TCB make effective use of hardware requires that a less primitive TCB subset make effective use of the protection-relevant features exported to it by the more primitive TCB subsets (e.g., execution domains, support for distinct address spaces) to separate those elements that are protection-critical from those that are not.

Statement from TCSEC

The TCB shall maintain a domain for its own execution that protects it from external interference or tampering.

Interpretation

If the TCB is composed of multiple TCB subsets, this requirement applies to each TCB subset.

Statement from TCSEC

The user interface to the TCB shall be completely defined and all elements of the TCB identified.

Interpretation

If the TCB is composed of multiple subsets, this requirement applies to the interface between the TCB subsets as well as the user interface to the whole TCB.

Statement from TCSEC

It shall make effective use of available hardware to separate those elements that are protection-critical from those that are not.

Interpretation

If the TCB is composed of multiple subsets, each TCB subset must make use of facilities provided to it by more primitive TCB subsets, such as support for execution domains and for distinct address spaces, to achieve the required separation.

A1-3.1.2 SYSTEM INTEGRITY

This requirement applies as stated in the TCSEC to every TCB subset that includes hardware or firmware. Any TCB subset that does not include hardware or firmware is exempt from this requirement.

A1-3.1.3 COVERT CHANNEL ANALYSIS

This requirement applies as stated in the TCSEC to the entire TCB. When the TCB is made up entirely of TCB subsets meeting the conditions for evaluation by parts, analysis of the individual TCB subsets satisfies this requirement. Otherwise, covert channel analysis of the entire TCB must be performed (even if the results of covert channel analysis of the individual TCB subsets were available).

A1-3.1.4 TRUSTED FACILITY MANAGEMENT

This requirement applies as stated in the TCSEC to the entire TCB. Any "operator" or "administrator" functions intrinsic to an individual TCB subset must be supported by that TCB subset or by a more primitive TCB subset.

A1-3.1.5 TRUSTED RECOVERY

This requirement applies as stated in the TCSEC to the entire TCB and to the individual TCB subsets. The cooperative recovery actions of the TCB subsets making up the TCB must provide trusted recovery for the overall TCB. Otherwise, trusted recovery evaluation must address the entire TCB (even if the individual TCB subsets meet the trusted recovery requirements).

A1-3.2 LIFE CYCLE ASSURANCE

A1-3.2.1 SECURITY TESTING

This requirement applies as stated in the TCSEC to the entire TCB. If a TCB consists of TCB subsets meeting the conditions for evaluation by parts, the satisfaction of the requirements by each TCB subset satisfies the requirement for the entire TCB. Otherwise, security testing of the entire TCB must be performed (even if the results of testing the individual TCB subsets were available).

A1-3.2.2 DESIGN SPECIFICATION AND VERIFICATION

This requirement applies as stated in the TCSEC to every TCB subset, with the following specific additional interpretations.

It must be demonstrated that the security policy enforced by the composite TCB is represented by the collection of policy models for the individual TCB subsets.

The argument that the descriptive top level specification (DTLS) and formal top level specification (FTLS) are consistent with the TCB interface applies to the entire TCB. There is required an explicit and convincing description of how the set of top level specifications (one for each TCB subset) describes the TCB interface in terms of exceptions, errors, and effects.

Statement from TCSEC

A formal model of the security policy supported by the TCB shall be maintained over the life cycle of the ADP system and demonstrated to be consistent with its axioms.

Interpretation

If the TCB is composed of multiple TCB subsets, this requirement applies to the security policy of each TCB subset. An informal argument that the set of policy models represents the "security policy supported by the [composite] TCB" must be provided.

Statement from TCSEC

A descriptive top-level specification (DTLS) of the TCB shall be maintained that completely and accurately describes the TCB in terms of exceptions, error messages, and effects.

Interpretation

If the TCB is composed of multiple TCB subsets, this requirement applies to the DTLS of each TCB subset. An informal argument that the set of DTLSs accurately describes the TCB must be provided.

If there is a multifaceted policy (e.g., both mandatory access control and discretionary access control policies) in a particular TCB subset, then all facets must be represented in the DTLS and in the TCB subset's model.

Statement from TCSEC

A formal top-level specification (FTLS) of the TCB shall be maintained that accurately describes the TCB in terms of exceptions, error messages, and effects.

Interpretation

If the TCB is composed of multiple TCB subsets, this requirement applies to the FTLS of each TCB subset. An informal argument that the set of FTLSs accurately describes the TCB must be provided.

If there is a multifaceted policy (e.g., both mandatory access control and discretionary access control policies) in a particular TCB subset, then all facets must be represented in the FTLS and in the TCB subset's model.

Statement from TCSEC

The FTLS shall be shown to be an accurate description of the TCB interface.

Interpretation

If the TCB is composed of multiple subsets, this requirement applies to the interface between the TCB subsets as well as to the user interface of the composite TCB. The TCB interface description shall include an explanation of how to describe the total TCB accurately, in the context of the multiple TCB subset DTLSs.

Statement from TCSEC

A convincing argument shall be given that the DTLS is consistent with the model.

Interpretation

If the TCB is composed of multiple TCB subsets, this requirement applies to individual TCB subsets. Enforcement of all policies must be shown to occur in all situations (e.g., state transitions) required by the formal security policy model. In the case of a discretionary access control policy, the presence of the access control check at all identified state transitions is the total of what is required for demonstrating consistency between the DTLS and the model. This may be verified by inspection of the DTLS (that is, by visually checking that exception checks required by the model are present in the DTLS). For the mandatory access control policy, the DTLS must be shown by a convincing argument to be consistent with the model.

Statement from TCSEC

. . . a combination of formal and informal techniques shall be used to show that the FTLS is consistent with the model.

Interpretation

If the TCB is composed of multiple TCB subsets, this requirement applies to individual TCB subsets. Enforcement of all policies must be shown to occur in all situations (e.g., state transitions) required by the formal security policy model at the required occasions. In the case of a discretionary access control policy, the presence of the access control check at all identified state transitions is the total of what is required for demonstrating consistency between the FTLS and the model. This may be verified by inspection of the FTLS (that is, by visually checking that exception checks required by the model are present in the FTLS). For the mandatory access control policy, the FTLS must be shown by a combination of formal and informal techniques to be consistent with the model.

A1-3.2.3 CONFIGURATION MANAGEMENT

This requirement applies as stated in the TCSEC to every TCB subset in the TCB, with the following additional interpretation.

Because subsets of the TCB may be developed independently, a single configuration management system may not be feasible. However, the combination of configuration management systems used to support all the TCB subsets must meet all the stated requirements. The information describing the interrelations between separate TCB subsets and separate security policy models falls into the category of "all documentation and code associated with the current version of the TCB" in the TCSEC requirements.

A1-3.2.4 TRUSTED DISTRIBUTION

This requirement applies as stated in the TCSEC to the entire TCB. It can be met by satisfying the requirements for each TCB subset if procedures exist for assuring that all TCB subsets upon which a particular TCB subset depends (that is, the more primitive TCB subsets) are exactly the same version as specified for the TCB subset in question.

A1-4 DOCUMENTATION

A1-4.1 SECURITY FEATURES USER'S GUIDE

This requirement applies as stated in the TCSEC to every TCB subset in the TCB. This collection of guides must include descriptions of every TCB subset in the TCB and explicit cross-references to other related user's guides to other TCB subsets, as required. In addition, interactions between mechanisms within different TCB subsets must be clearly described.

A1-4.2 TRUSTED FACILITY MANUAL

This requirement applies as stated in the TCSEC to the TCB and to every TCB subset in the TCB.

This requirement can be met by providing a set of manuals, one for each distinct (non-replicated) TCB subset. Each manual shall address the functions and privileges to be controlled for the associated TCB subset. Additionally, it must clearly show the interfaces to, and the interaction with, more primitive TCB subsets. The manual for each TCB subset shall identify the functions and privileges of the TCB subsets on which the associated TCB subset depends. Also, the TCB subset's manual shall identify which, if any, configuration options of the more primitive TCB subsets are to be used for the composite TCB to operate securely.

Any pre-defined roles supported (e.g., database administrator) by the TCB subset shall be addressed.

The means for correlating multiple audit logs and synonymous user identifications from multiple TCB subsets (if such exist) shall also be addressed.

The trusted facility manual shall describe the composite TCB so that the interactions among the TCB subsets can be readily determined. Other manuals may be referenced in this determination. The manuals that address the distinct modules of the TCB and the generation of the TCB need to be integrated with the other trusted facility manuals only to the extent that they are affected by the use and operation (versus the development) of the composite TCB.

The TCB modules that contain the reference validation mechanism must be associated with the TCB subset to which they belong. The procedure for generating a new TCB after modifying only one (or several) TCB subsets must be described. This may be accommodated by independent regeneration of the individual TCB subsets or by regeneration of only the affected TCB subsets.

A1-4.3 TEST DOCUMENTATION

This requirement applies as stated in the TCSEC to the composite TCB.

A1-4.4 DESIGN DOCUMENTATION

This requirement applies as stated in the TCSEC to the composite TCB, with the following specific additional interpretations:

Requirements concerning models, FTLS and DTLS, apply to individual TCB subsets.

The requirement concerning the description of interfaces between modules of the TCB includes the interfaces between TCB subsets.

The documentation of the implementation of a reference validation mechanism must include justification for meeting the conditions for evaluation by parts.

The A1 requirement to describe clearly non-FTLS internals of the TCB applies to TCB subsets.

Statement from TCSEC

The interfaces between the TCB modules shall be described.

Interpretation

If the TCB is composed of multiple subsets, this requirement applies to each TCB subset and the interfaces between TCB subsets.

Statement from TCSEC

The specific TCB protection mechanisms shall be identified and an explanation given to show that they satisfy the model.

Interpretation

If the TCB is composed of multiple subsets, this requirement applies to each TCB subset and shall include the protection mechanisms which support the conditions for TCB subset structure and separate subset-domains.

Statement from TCSEC

The descriptive top-level specification (DTLS) shall be shown to be an accurate description of the TCB interface.

Interpretation

If the TCB is composed of multiple subsets, this requirement applies to the interface between the TCB subsets as well as to the user interface of the composite TCB. The TCB interface description shall include an explanation of how to describe the total TCB accurately, in the context of the multiple TCB subset DTLSs.

Statement from TCSEC

Documentation shall describe how the TCB implements the reference monitor concept and give an explanation of why it is tamper resistant, cannot be bypassed, and is correctly implemented.

Interpretation

If the TCB is composed of multiple TCB subsets, this requirement is interpreted to apply to each TCB subset with respect to its specific technical policy. In addition, there must be documented an informal argument that the cooperative action of the TCB subsets makes the TCB itself tamper resistant, non-bypassable, and correct.

Statement from TCSEC

The TCB implementation shall be informally shown to be consistent with the DTLS.

Interpretation

If the TCB is composed of multiple TCB subsets, this requirement is interpreted to apply to individual TCB subsets.

Statement from TCSEC

The TCB implementation shall be informally shown to be consistent with the FTLS.

Interpretation

If the TCB is composed of multiple TCB subsets, this requirement is interpreted to apply to individual TCB subsets.

Statement from TCSEC

Documentation shall describe how the TCB is structured to facilitate testing and to enforce least privilege.

Interpretation

If the TCB is composed of multiple subsets, this requirement is interpreted to apply to individual TCB subsets as well as to the overall TCB.

APPENDIX B

GENERAL ITEMS

1. PERSPECTIVE ON ASSURANCE

This Trusted Database Management System Interpretation (TDI) of the Trusted Computer System Evaluation Criteria (TCSEC) derives its perspective on assurance directly from the reference monitor concept as recorded in the Anderson Report [1] and as embodied in the TCSEC. In both the reference monitor concept and in the TCSEC, the assessment of a system for trust characteristics involves a single, global review of the system at issue. That same perspective of an even, global review of a candidate trusted database management system (DBMS) is present in the TDI, in that only complete systems will be considered for assessment. That is, neither software packages in isolation nor systems that satisfy only a subset of the TCSEC. requirements will be considered for evaluation or accreditation. The interpretation of requirements, both in Part 1, "Technical Context," and Part 2, "Interpreted Requirements," is consistent with both community experience in designing and assessing trusted systems, and the precedents of the reference monitor concept and the TCSEC. The interpretations, therefore, provide special guidance for the task of evaluating (or accrediting) candidate systems composed of distinguishable parts. However, the interpretations neither attenuate nor delete TCSEC requirements.

It is worth noting that the introduction of the TCSEC with its metric for the evaluation of whole systems had as one goal the simplification of the task of accrediting computer systems for use in the processing of sensitive information. The evaluation process was not intended to replace the accreditation process but to provide input to that process. It must be recognized that there will be occasions when a fielded suite of computer systems, each evaluated against the TCSEC requirements at a particular class, will not be able to be assigned a TCSEC class, nor is it necessary to be able to make such an assignment. The accreditation decision includes the assessment of risk of a particular system configuration in a particular environment; a decision to connect a suite of TCSEC evaluated systems may have to be made without a uniformly applied TCSEC-like assessment over the entire system.

2. PROCUREMENT OPTIONS

The Trusted Computing System Evaluation Criteria (TCSEC) and its published interpretations, including this Trusted Database Management System Interpretation, have as a primary purpose the "provision [of] a basis for specifying security requirements in acquisition specifications." [8, p. 2] In the area of trusted DBMS and trusted systems that include database management functionality, there is a range of options open to an acquisition organization. These options need to be understood by the acquisition managers and their staffs to allow them the greatest possible flexibility in matching operational requirements with a combination of available products and the state of the art in system integration and development.

The fundamental point is the distinction between the target trust class (that is, C1, C2, B1, B2, B3, or A1) needed for a particular installation and the degree of trusted DBMS functionality that is required. Succinctly, a system that requires a particular level of trust (B2, for example) and DBMS functionality does not necessarily require an evaluated trusted DBMS at the level of B2. In fact, if the statement of required capability allows it, meeting the requirement without a trusted DBMS could well be the preferred risk-reducing approach. This is generally true because the more sophisticated the trusted DBMS requirement, the more likely it is that the current vendor base will not be able to supply the needed functionality (with the requisite assurance) from the normal product line. Further, even if one can specify carefully just what additional assured capability is needed so that a program-specific development can be undertaken, the lack of community experience and consensus on advanced trusted DBMS issues and implementations introduces the potential for significant programmatic, schedule, and cost risks.

Although a full description of options for the acquisition manager is beyond the scope of this document, a representative description of some of the options that could be considered is included below. The options include (1) multiple copies of a DBMS running at different levels, each maintaining single-level databases; (2) a single copy of a DBMS, but with each database maintained at a single sensitivity level (i.e., no sharing of data between databases); (3) a single copy supporting single level databases, but with limited sharing, perhaps of a "snapshot" nature; and (4) DBMSs that allow databases that include data of several sensitivity levels. This option admits of subcases from the very simple to the very complex.

To illustrate the options listed above, consider a command post where a commander's staff uses a single computer system. Included on the staff are logistics, weather, and intelligence organizations, each dealing with information of differing (maximum) sensitivity. If all three organizations require similar DBMS functions, there might be a variety of ways to provide that functionality.

(1) If the product DBMS-1 suited the needs of each of the organizations and there were no requirement to share data between them, then three copies of DBMS-1 could be used, running at, for example, TOP SECRET, SECRET, and CONFIDENTIAL, respectively, and maintaining separate single-level databases. If the organizational missions do not require multilevel operations or sharing, this option could be perfectly suited to the operational need. In this case, every copy of DBMS-1 would be running as an application outside the TCB and would not have to be evaluated at all, neither as a product nor as a developed system. The advantages of this option are that commercial, off-the-shelf systems can be used (both the DBMS and the underlying trusted operating system) and no evaluation risk is entailed. The disadvantages are the limited flexibility and the hidden fact that the installation procedures for many DBMSs require the insertion of code into the heart of the underlying computer system, insertions that would un dermine the evaluation rating and thus the confidence in the trusted operating system.

(2) A cost advantage could be realized in the preceding case if there were a product, DBMS-2, such that a single copy could provide DBMS functionality at all three levels. This capability requires that the base trusted operating system and the DBMS itself are designed so that the DBMS code uses scratch space to allow the code to be shared without the potential for mixing control or metadata in workspaces, indices, and stacks. Not all commercial DBMSs have this property, so this option will be less available than case (1). Note that in this case also, the databases themselves are single-level and the workspace used by the DBMS itself will have to be single-level.

(3) If the operational requirements are that the intelligence and logistics organizations must have access to the weather data maintained by the weather organization, the simplest technical solution would be to periodically provide a snapshot of the needed weather data for use by the other organizations. The database management system DBMS-2 above could provide a

solution in this case if a portion of the weather database could be copied onto diskette (or even into another file) for the other organizations to incorporate into their own DBMS operations. The disadvantage of this approach is that the information will not be as current as that available to the weather organization itself. Frequently, however, that lack of currency may be a reasonable price to pay for the enormous reductions in cost and risk in procurement and maintenance.

(4) If the operational requirements will not allow anything less than real-time sharing of information, then DBMS-2 will not be sufficient. At this point, the operational requirements have forced the inclusion of the most sophisticated solution to a trusted system with DBMS requirements, a true multilevel DBMS. In this case, it remains a valuable goal to minimize the complexity of the needed sharing. If the DBMS is providing a functionality that looks like tables to the user, then it is less complex if each table is at a single level than if each row (or each column) could be at a different sensitivity level. The most complex situation is when each entry in the table could be at a different sensitivity level. During the procurement and development process, it would be worthwhile to structure the procurement to favor solutions that are as simple as possible from a multilevel trusted DBMS standpoint.

3. ALTERATION OF PREVIOUSLY EVALUATED TCB

The discussion in Part 1, "Technical Context" arose from consideration of the conditions under which it is possible to add an application layer, such as a DBMS, to another TCB in such a way as to allow the system rating to be determined by evaluating the system elements (i.e., the subsets) separately. The benefit to both the applications vendor and the evaluators derives from the fact that the DBMS operates as an untrusted subject relative to the underlying TCB (even though the DBMS enforces its own policy). Therefore, there is no need to re-examine previous evaluation evidence; any and all actions performed by the application layer are completely constrained in accordance with the security policy defined for the underlying TCB.

The savings in evaluation effort is predicated on the assumption that the vendor of the application layer makes no changes of any kind to the other TCB. In effect, the argument made by the vendor is as follows:

(a) The underlying TCB enforces policy P[i]. The validity of the claims about the underlying TCB has already been established, and these claims remain valid because the underlying TCB has not been altered in any way and because the DBMS is completely constrained by that policy (i.e., P[i] cannot be violated by any action of the DBMS).

(b) The application is a TCB subset which enforces policy P[k].

(c) Thus, the policy enforced by the composite system (i.e., the policy enforced at the user interface of the composite TCB) is merely a combination of the policies of the individual TCB subsets.

Because there is neither theory nor experience which allows one to make general, a priori statements about the effects of arbitrary changes, any alterations to a previously evaluated product must, in general, be considered to result in a product whose security attributes, and thus whose rating, is unknown. Thus, if the previously evaluated product is altered, argument (a) cannot be made merely by referencing the published evaluation report. It becomes the responsibility of the DBMS vendor to state P[i] (i.e., identify the policy enforced by the altered product) and to demonstrate that the implementation satisfies the appropriate TCSEC requirements. Hence, at least some evaluation evidence focused on the underlying TCB must be provided by the vendor of the application layer. The amount of evidence required will be determined by the type, extent, and amount of change, as well as the characteristics of the original TCB.

This is not to say, however, that changes always invalidate all previous evaluation evidence. Rather, that there is no way to predict what effect arbitrary change will have on that evidence. Clearly, some changes will invalidate a substantial part, if not all, of the previous evaluation results, requiring a completely new evaluation. In some cases it will be virtually impossible to determine the full effect of even relatively simple changes, whereas in other instances it may be possible to determine the effects of at least some changes quite precisely. In a well-modularized system, changes to the internals of a module might be shown to have no functional or security effect outside of the module. Even changes to the module that alter its interface might be shown to have effects which do not propagate beyond those modules which have a direct interface to the altered module.

In either case however, at least some evidence must be produced about the underlying, altered TCB. Thus, the vendor who alters the product which is hosting his application becomes responsible for any and all evidence required to substantiate the claims being made for both the application and the underlying TCB.

In fact, it is always the case that the DBMS vendor is responsible for all the evidence required to demonstrate that the system (i.e., the trusted components of the application plus the underlying TCB) has the level of trust claimed for it. In the case of strict subsetting for evaluation by parts, in which the application is layered onto an unaltered, previously-evaluated TCB, part of the evidence is satisfied by referencing the previous evaluation results and the commercially-available specifications for that portion of the system. However, if the previously evaluated TCB has been altered, the applications vendor is now responsible for demonstrating that the underlying TCB has the level of trust being claimed for it. How much, if any, of the previous evaluation evidence is still valid is a matter to be resolved.

The development of the subsetting notion has been motivated by the need for evaluating systems whose elements may have been developed by different vendors. Consequently, the discussion of TCB subsets has been largely focused on the topic of extending the policy enforcement attributes of previously evaluated TCBs. However, the notion of TCB subsetting provides a perfectly general design method. A TCB can be structured and policy enforcement allocated to simplify both analysis and evaluation. To the extent that each TCB subset in a subsetted system satisfies the conditions of TC-4.3, the evaluation can be factored along policy lines, and a rating for the composite system determined.

4. SATISFYING RVM REQUIREMENTS

The evaluation of a system whose TCB is made up of a set of TCB subsets follows a logical flow that makes it an orderly process. The initial step is satisfying the conditions for evaluation by parts. Those conditions are as follows:

Identification of the candidate TCB subsets;

Allocation of the policy (the clear statement of the technical policies enforced by the individual TCB subsets, stated in terms of the subjects and objects for that TCB subset);

Demonstration that each candidate TCB subset contains its own trusted subjects;

Specification of the structure of the TCB subsets (as a collection of abstract machines);

Identification of sets of domains (called "subset-domains") assigned for the execution of the individual TCB subsets; and

Identification of what externally stated properties of TCB subsets will be used to argue convincingly and independently for the RVM nature of each of the constituent TCB subsets.

The successful completion of this step, especially the "support for RVM arguments" will result in a conditional approval of two items: a set of goals in the evaluation of the more primitive TCB subsets and the feasibility of being able to argue the RVM properties of less primitive TCB subsets using no more information about the more primitive TCB subsets than the identified goals. The goals for the more primitive TCB subsets will be a set of mechanisms, characteristics, or properties whose proper, assured functioning will have to be demonstrated. Examples are domain mechanisms, mandatory integrity policy enforcement mechanisms, and a special category of object with associated content-preservation guarantees. These mechanisms or characteristics or properties are strictly a function of the more primitive TCB subset and will have to be evaluated and approved in a (possibly) later part of the evaluation process. The conditional approval of the feasibility of constructing an independent RVM argument for less primitive TCB subsets relies on an interplay between the proposed mechanisms of the more primitive TCB subset and the anticipated needs of the RVM argument for the less primitive TCB subset.

The next steps of the evaluation process, with regards to the reference validation mechanism requirements, involve the independent evaluation of the TCB subsets. TCB subsets that have been identified as providing features or mechanisms on which other TCB subsets will rely for RVM arguments will have to be demonstrated to provide the stated mechanisms with the same level of assurance as the target evaluation class of the entire system. If all the identified mechanisms can be validated, the conditional approval of the "support for RVM arguments" condition remains unchanged. If some mechanism cannot be properly established from either a functional or an assurance perspective, then the conditional approval of that portion of the "support" condition is withdrawn and the evaluation effort must regroup.

TCB subsets that were projected to be able to complete RVM arguments successfully using no more than the identified "support" mechanisms and features will have to have full RVM arguments advanced and accepted by the evaluation team. There are three possible outcomes: (1) it could be shown that the TCB subset in question does not meet the RVM requirements; (2) it could be shown, using the external descriptions and assurances of the mechanisms of the more primitive TCB subsets, that the less primitive TCB subset does meet the RVM requirements; or (3) it might be impossible to determine whether or not the TCB subset meets the requirements. In case (1), the TCB subset failed to meet its reference validation mechanism requirements and the design team must regroup. In case (2), the TCB subset satisfies its reference validation mechanism requirements. In case (3), the conditional approval of the "support for RVM arguments" condition will be withdrawn and the design and evaluation teams will have to agree on an alternate course of action.

In the case that an attempt to establish RVM properties for a less primitive TCB subset could not be completed (case (3) above), it might well be that additional mechanisms or features of the more primitive TCB subset would allow the RVM arguments to be completed successfully. In that case, the evaluation team and the design team would have to establish a new, augmented set of mechanisms for the "support" condition. The evaluation could then continue with the new mechanism requiring validation by the evaluation team and the argument for the RVM properties of the less primitive TCB subset having to be re-attempted.

It is important to note that the dependency of the less primitive TCB subsets on the assured policies and RVM supporting mechanisms makes it imperative that the evaluation of the whole TCB be done by a single evaluation team through the final determination that the system complies with the full set of requirements for the target class. Thus, in particular, an evaluation team addressing an evaluation by parts (including the case of a TCB subset that has been previously evaluated and placed on the EPL) must be kept together for the entire evaluation. Local evaluation of one TCB subset does not justify dissolving the responsible subteam. Later results, global or local to another TCB subset, could require a full evaluation team current on all aspects of the evaluated configuration. This does not mean, of course, that the original evaluation team for a previously evaluated EPL product has to be reassembled. A new team, part of which may be delegated prime responsibility for that TCB subset, suffices, as long as the full team is kept together for the whole evaluation.

5. SUBSET DEPENDENCY

For candidate TCB subset M, sM denotes the specification of M, including as a minimum, the statement of the technical policy P of M. The term vM denotes the (engineering) demonstrations of the correctness of the implementation of M with respect to its specification. A (candidate) TCB subset A "depends (for its correctness)" on (candidate) TCB subset B if and only if the arguments within vA assume the correctness of the implementation of B with respect to sB.

In less precise terms, the specification sM defines what M is supposed to do. M does whatever its implementation allows, features and all. How well M does compared to what sM says it should do is encompassed in the engineering arguments vM. If, in the argument vA, one has to assume that all or part of sB accurately describes what B does, A "depends" on B for its correctness.

Example 1: Use of Provided Objects

Suppose TCB subset B provides "file" as a mediated object under its technical policy P[B] and that candidate TCB subset A uses B-files to store data and executables. If vA uses the fact that different B-files are distinct and access to them is constrained by the technical policy P[B] (assumptions that are nearly certain to apply), then vA is relying on the fact that sB and B match and in this case, A depends on B.

Example 2: Mutually Suspicious Systems

Suppose A and B are mutually suspicious airline reservation systems hosted in two interconnected systems belonging to two distinct organizations. It is assumed that sA and sB both provide for a capability to accept reservations over the network from "foreign" systems using a mutually defined protocol. The protocol is carefully and completely specified (within both sA and sB) and both vA and vB demonstrate, to the desired level of satisfaction, that A and B are correct with respect to sA and sB, respectively. A does not depend for its correctness on B because sA is complete: for whatever sequence of bits it receives from B, sA specifies exactly what behavior A must exhibit, and vA demonstrates that it does exhibit that behavior. Similarly, B does not depend upon A for its correctness. Neither A nor B depends on the other.

There may well exist a "system specification" that specifies the desired behavior of the entire system, but that is irrelevant to the arguments that A and B are individually correct with respect to their own specifications. It may even be the case that both A and B are individually correct, while the combined system is incorrect with respect to the "system specification." That is, two correct subsystems can be combined improperly with respect to the desired "system specification."

Example 3: Use of Remotely Provided Functionality

Suppose A is a mail server and B a generic SQL DBMS. The specification sA (as might be expected) makes no mention of a DBMS; it simply specifies the interface behavior (to its human clients) of the mail system. In vA, however, we find that the software for A uses tables provided by B to store messages. A and B are on separate, interconnected machines. Neither sB nor vB make mention of the mail system at all. As in the preceding example, sB completely specifies the behavior of B for all received bit patterns and sequences. Here, A depends upon B, but B does not depend on A. Note that data flow in both directions is completely legitimate and does not compromise in any way the "integrity" (correctness) of B. Dependency is distinct from "data flow."

This example shows that a superficial examination of the "architecture" of a domain structure is insufficient to determine which candidate TCB subsets depend upon other TCB subsets. Superficially, this architecture is the same as the example of mutually suspicious systems above, but here A depends on B. It also shows that an examination of the interface specifications is insufficient. Finally, it shows that dependency is not the same as the notion of "privilege" (as normally understood in the context of operating systems to mean loosened restrictions on allowed functions and accesses), because there is in this example no sense in which B has access to all of A's internal structures. It only has access to some of them.

Example 4: Use of Locally Provided Functionality

Suppose A and B are the mail server and SQL DBMS of the preceding example, except that A is implemented in a less privileged ring than B. That is, the interconnect is replaced by a ring-crossing service call. Obviously, A still depends on B and B does not depend on A. The change is that now B has potential access to any of A's structures. The general rule for the use of domain protection mechanisms (such as rings) is that if B is privileged with respect to A, then A necessarily depends on B (simply by virtue of B's privilege to access any of A's structures). Thus, a detailed examination of sA and vA is unnecessary to establish dependency.

Cautionary Example

Suppose that A and B are "mutually dependent"; that is, A depends on B and B depends on A. This means that vA assumes that B is implemented correctly with respect to sB, and vB assumes that A is implemented correctly with respect to sA. Further suppose that both vA and vB are valid (reasonable and compelling). One would hope that it follows from this that both A and B are correct. Unfortunately, this is not always the case.

If A and B are both correct with respect to sA and sB, then vA is a valid argument with a true premise and is therefore true. The same is true for B and vB.

Suppose, however, that A is implemented incorrectly with respect to sA and B is implemented incorrectly with respect to sB. vA is a valid argument with a false premise; it is thus valid but (possibly) untrue. Similarly, vB is valid but (possibly) untrue. This case shows that it is quite possible for vA and vB to both be valid while A and B are both (individually) incorrectly implemented.

What has been exposed here is a hidden case of circular reasoning: the argument that A is correct is based on the assumption that B is correct, and the argument that B is correct is based on the assumption that A is correct. Thus, vA depends (circularly) on the assumption that A is correct, and vA reduces to the following tautology:

if vA is correct with respect to sA then vA is correct with respect to sA.

It is thus possible in principle for mutually dependent subsystems A and B to have vA and vB to be logically valid while either A or B, or both, are incorrect with respect to their specifications (sA or sB).

To make this result concrete, consider two airline reservation systems, A and B, based on the mutually suspicious systems of example 2 above. Suppose that A maintains information about all flights originating or terminating in the United States while B maintains information about flights originating or terminating in Europe. Assume sA includes a statement that A will generate flight itineraries from an origin to a destination, with appropriate provision for connections. "Appropriate provision for connections" means allowing enough time to change planes without requiring passengers to wait an excessive period of time. Further assume that sB includes a similar statement. From the assumption that A meets sA and B meets sB, one can construct a valid argument that A meets its specification sA for itineraries orginating or terminating in either the U.S. or Europe. A similarly valid argument can be made about B. If, however, A stores flight segment times in the local time of the airport and B stores them i n Greenwich Mean Time, an itinerary generated by either A or B that relies on information from the other will be incorrect because of the time differences. Thus, A will not generate accurate, timely flight itineraries, even though a valid argument that it does can be constructed. This difficulty arises from the presumption that A and B are mutually dependent.

6. TAMPER RESISTANCE ARGUMENTS

The requirement to demonstrate that individual TCB subsets exhibit the reference validation mechanism tamper resistance property deserves special attention. In a subsetted architecture there are two (related) aspects to this requirement. The first is the ability of a less primitive TCB subset to maintain control over access to the objects that implement its logic and data structures. The second is establishing the assurance that policy-critical or correctness-critical data used by a TCB subset is persistent (that is, that it does not change or become contaminated with other data between the execution of instructions).

A less primitive TCB subset will be using objects and subjects provided by a more primitive TCB subset. Thus, policy-critical data will be stored in objects that are provided by the more primitive TCB subset rather than in some system entity created and maintained by the less primitive TCB subset itself. One part of the tamper resistance argument focuses on being able to demonstrate that no alteration of either the policy-critical data or of the TCB subset's code is possible. That is, no system entity external to a TCB subset (with the possible exception of more primitive TCB subsets upon which it depends) should be capable of causing arbitrary changes to that TCB subset's code or data structures. If a third, not more primitive TCB subset (or a subject totally outside the TCB) were able to gain access to an object where policy-critical data was stored, the tamper resistance property could not be established for the TCB subset in question. For a most-primitive TCB subset, a wide variety of techniques could be used to limit access to an object in which such policy-critical data is stored (e.g., prohibition on the sharing of objects, strict ownership control of the ability to extend access permission, and mandatory access controls). Similarly, the conditions for evaluation by parts require that the system designer identify a set of mechanisms and their assured properties as the basis for demonstrating tamper resistance for each TCB subset.

The second topic is the assurance that the contents of the objects that store a TCB subset's policy-critical data not change except through the execution of that TCB subset's logic. If a data structure that identified an exported object (such as tables or tuples or entities) were to have extraneous data inserted by a more primitive TCB subset (for example, as a result of garbage collection or the random action of memory management), then no basis would exist for arguments concerning the correct implementation of the less primitive TCB subset. For a most primitive TCB subset (which includes supporting hardware), the argument that the policy-critical data is kept current and correct can be made strictly on the basis of that TCB subset. However, when a TCB subset obtains resources from a more primitive TCB subset, the argument cannot be made strictly on the basis of the design of the TCB subset. Rather, the argument must proceed from assured mechanisms provided by more primitive TCB subsets. This is exactl y analogous to the case of a reference validation mechanism for which one relies on mechanisms not strictly included in the design of the policy-enforcing elements to establish the requisite properties of non-bypassability and tamper resistance. A variety of mechanisms might provide a basis for demonstrating that the implementation of a TCB subset is correct, even though resources are obtained from a more primitive TCB subset (e.g., type-enforcement).

7. RATIONALE FOR LOCAL AND GLOBAL REQUIREMENTS

Section TC-5, "General Interpreted Requirements," lists the requirements of the TCSEC according to how the requirements apply to a TCB made up of more than one TCB subset. The general rationale for distinguishing which requirements can be satisfied by independent analysis and consideration of the TCB subsets was to consider the requirements one at a time to determine whether satisfaction of the requirement by the individual TCB subsets would necessarily mean that the entire system satisfies the stated requirement. For some requirements, such as those for certain documentation, it is clear that the requirement is factorable; that is, it is satisfied for the composite TCB if it is satisfied for each of the TCB subsets individually. For other requirements, individual system characteristics could make factorability of a requirement patently obvious, but not all systems would enjoy such simple factorability. An example would be trusted path requirements for security-related events: if all security-related ev ents occur in the most primitive TCB subset, satisfaction of the requirement by that TCB subset suffices to demonstrate that the system meets the requirement, but for systems that have security-relevant events in less primitive TCB subsets, some explanation of the cooperative action of the TCB subsets would be required. For still other requirements, one can expect that the satisfaction of the requirement for the entire system will always require some explanation of the cooperative action of the constituent TCB subsets. Provision of a coherent audit record across events in several TCB subsets is in this category.

In the paragraphs below, a brief rationale for identifying requirements as wholly or partially global is provided. Those requirements that are not listed are considered to be completely local.

Subject Sensitivity Labels

At level B2 and above, the TCSEC requires the following:

The TCB shall immediately notify a terminal user of each change in the security level associated with that user during an interactive session. A terminal user shall be able to query the TCB as desired for a display of the subject's complete sensitivity level.

For subsetted architectures, the user interface could be to a TCB subset that does not support a mandatory access control policy. Thus, a change noted by a more primitive TCB subset that does support such a policy would have to be relayed to the user, possibly through cooperative action of the full set of TCB subsets. Similarly, a request by a terminal user for the complete sensitivity level could be initially received by a TCB subset that does not support a mandatory access control policy and will require cooperation between TCB subsets to determine the complete subject sensitivity level and to provide that information to the requesting user.

Identification and Authentication

The identification and authentication requirements in the TCSEC address the need to correctly associate authorizations with subjects. In a TCB made of several TCB subsets, it is possible that only one of the TCB subsets will provide identification and authentication, which will be used by all the less primitive TCB subsets. Alternatively, identification and authentication may be provided directly in more than one TCB subset. In either case, the TCB subsets have to work cooperatively to use identification and authentication data for uniquely identifying users and for associating users with auditable actions.

Trusted Path

At B2, the only required uses of trusted path are login and authentication. At B3 and above, occasions "when a positive TCB-to-user connection is required (e.g., login, change subject security level)" are included. In both cases, a system vendor may choose to use trusted path for situations where the security-relevant event could be recognized or handled in more than one TCB subset. On those occasions, the careful coordination of all the involved TCB subsets in the correct handling of trusted path situations must be shown. If a single TCB subset implements trusted path and all the invocations of trusted path are limited to that TCB subset (that is, the flow of control in responding to a trusted path initiation never leaves the TCB subset until the response is completed), then nothing further would be required. The description of the limitation of trusted path to a single TCB subset will suffice for the global part of the requirement, leaving only the demonstration of local satisfaction of the requireme nt by the identified TCB subset.

Audit

If each of several TCB subsets meets the audit requirements locally, then there is the issue of whether the set of audit records meets the requirements of being able to note and record individual user actions, and at B3 and above, to be able to initiate required action. If not all the TCB subsets meet the audit requirements locally, then the requirements must be satisfied by the cooperative action of the set of TCB subsets. In both cases, consideration of the audit characteristics of all the TCB subsets has to be part of determining that the entire TCB meets the audit requirements.

System Architecture

For many of the system architecture requirements, demonstrating that a requirement is satisfied by all of the consitituent TCB subsets is sufficient to demonstrate that it is satisfied for the composite TCB. The requirements for the "TCB [to] maintain a domain for its execution" and for the TCB to "maintain process isolation through the provision of distinct address spaces" could be satisfied by the entire TCB without each constituent TCB meeting the requirement.

The requirement for the TCB to maintain a domain for its execution implies the need for each TCB subset to have a domain for its own execution, isolated from both untrusted portions of the system and from less primitive TCB subsets. The exact wording of the TCSEC requirement could be read as disallowing a less primitive TCB subset from occupying a domain provided by a more primitive TCB subset and to allocate its subjects to domains that do not have access to its own domain: the verb "maintain" could be (erroneously) read as requiring each TCB subset to create and maintain its own domain for execution. The proper interpretation is that the TCB as a whole must provide and maintain such domains for execution, rather than each individual TCB subset.

Similarly, the exact wording of the TCSEC requirement on the "maintain[ing] of process isolation through the provision of distinct address spaces" could be read as requiring each TCB subset to provide distinct address spaces. The proper interpretation is that the TCB as a whole must provide and maintain process isolation, either through provision and subsequent use of distinct address spaces, or through provision of distinct address spaces by every TCB subset.

Covert Channel Analysis

This requirement applies as stated in the TCSEC to the entire TCB. When the TCB is made up entirely of TCB subsets meeting the conditions for evaluation by parts, analysis of the individual TCB subsets suffices to satisfy this analysis requirement. Otherwise, covert channel analysis must address the entire TCB (even if the result of covert channel analyses of the individual TCB subsets were available).

Trusted Facility Management

The ability to run a trusted system facility properly applies to the combination of TCB subsets making up the TCB. This requirement should not be difficult to meet, provided that the individual TCB subsets meet the requirement and the interactions between the TCB subsets at the facility management level are clear.

Trusted Recovery

In the case of "an ADP system failure or other discontinuity," each TCB subset in a B3 or above system needs to be able to recover "without a protection compromise." Further, the recovery actions of distinct TCB subsets needs to be coordinated and combined so that the resulting system is not only recovered as far as each TCB subset is concerned, but is also recovered as a composite TCB.

Security Testing

This requirement applies as stated in the TCSEC to the entire TCB. If a TCB consists of TCB subsets meeting the conditions for evaluation by parts, the satisfaction of the requirements by each TCB subset suffices to satisfy the requirement for the entire TCB. Otherwise, security testing must include testing of the entire TCB (even if the results of testing the individual TCB subsets are available).

Design Specification and Verification

For many of the design specification and verification requirements, demonstrating that a requirement is satisfied by all of the consitituent TCB subsets is sufficient to demonstrate that it is satisfied for the composite TCB. The requirements for a "formal model of the security policy supported by the TCB" and that the DTLS at B3 and the FTLS at A1 "be an accurate description of the TCB interface" apply in a limited way to the entire TCB.

After complying security models are provided for the individual TCB subsets, a convincing argument is required to explain how the set of models represents abstractly the policy of the entire system.

After complying top-level specifications (DTLS at B3 and FTLS at A1) are provided for the individual TCB subsets, an explicit and convincing description of how the set of top-level specifications describe the TCB interface with respect to exceptions, errors and effects must also be provided.

8. CONTENT-DEPENDENT AND CONTEXT-DEPENDENT ACCESS CONTROL

An attractive proposition in a database managment system is to implement access controls that depend in some way on the values of the data in a storage object or the context in which the information is accessed. For example, one might desire to limit access to all personnel records in a database according to the salary value (content-dependent access rules). On the other hand, a company's earnings report might be held in confidence until announced at the stockholders' meeting, at which time it is public information (context-dependent access rules).

This document does not encourage or endorse mandatory access control on storage objects that are based on the content of data values or on the context in which the information is viewed. Given that these are research topics, it is prudent to take this conservative stance. Research and current practice are insufficient to allow definitive guidance on such implementations.

9. BULK LOADING OF A DATABASE

The bulk loading of a database into (perhaps thousands of) database objects must be done with care. If the data to be loaded are unlabeled, it may not be practical to require an authorized user to examine the data to be loaded into each object and assign it a sensitivity label. Instead it may be more practical to assign labels to the data according to the sensitivity label of the single-level device that is used to import it. In this way, bulk loading may be done in single-level stages.

Even importing labeled multilevel data may prove difficult. The imported data records may be organized on the input device in accordance with their logical structure, not their sensitivity levels. For some trusted DBMS architectures, this requires that the TCB first separate the data by sensitivity levels and subsequently load the data into the database as single-level structures.

Another problem with bulk loading of labeled data is granularity. It may be that the labeling granularity of data on the input device is different from the labeling granularity supported by the receiving trusted DBMS. As an example, the data being imported may be labeled at the file or field level, and the importing DBMS may support labeling at the tuple level. In such instances, the data would have to be mapped into objects of the proper labeling granularity as the data are being imported.

10. LOCAL ANALYSIS IN SYSTEM ASSESSMENT

The ability to distinguish and separately reference the results of local analysis of TCB subsets is a primary aspect of evaluation by parts, and the benefits which accrue are apparent in two, closely related, cases that arise in evalutions by parts. These may be thought of as dealing with the problems of "hosting" and "porting" although they are actually two aspects of the same problemthat of assessing a trusted system constructed of previously evaluated parts.

For the first case (i.e., that of "hosting"), consider an operating system which has been evaluated against the TCSEC requirements and has received a rating. Thus, the operating system is a TCB for which both the local and global analysis has been done. The results of the local analysis can now be used to support the evaluation of a TCB made up of the operating system (or, the more primitive TCB subset) and any proposed TCB extension (or, less primitive TCB subset). Suppose, for example, that vendor A chooses the rated operating system as the host for a DBMS product, which implements an access control policy. As described in TC-6, it is now possible, under the correct conditions, to re-use the results of the local analysis of the host operating system in developing a rating for the composite system. Even for those cases not meeting all the conditions for evaluation by parts, it may be possible that some, if not most, of the previous results are still valid. If vendor B also chooses the rated operating system as the host for his DBMS product, it will be possible (again, under the proper conditions) to develop a rating for the (new) composite system without having to repeat the local analysis of the host operating system. As an alternate case, suppose a site has need of an electronic mail service as well as the DBMS utility. The mail utility will operate in a peer-entity relation with the trusted DBMS utility (i.e., both the mail service and the DBMS depend on the host operating system, but neither depends on the other). Again, having the results of the local analysis of the host operating system eases the burden of assessing the security characteristics of the user interface to the composite system made up of the mail system and the host operating system. In short, the ability to distinguish and separately reference the results of the local analysis of the host operating system makes it feasible to evaluate the effect of adding arbitrary trusted applications, only by performing the local analyis for the application and any global analysis required.

For the second case, (i.e., that of "porting") the question becomes that of determining the effect of moving a known trusted application, such as a DBMS, across arbitrary host systems. Assume that a trusted DBMS product meeting the conditions for evaluation by parts has been evaluated on some trusted host, and a rating determined for the composite system. Clearly, the results of the local analysis of the trusted application available are also applicable to the analysis of a composite system made up of the trusted application and a different host operating system. Thus, having the local analysis of the trusted application will ease the evaluation burden associated with porting of trusted applications to different hosts. To the extent that the conditions for evaluations by parts have been satisfied, the local analysis of the application is valid by reference. Hence only the local analysis of the host operating system and the requisite global analysis is needed to assess the security attributes of the new composite system.

11. RATING MORE COMPLEX SYSTEMS

The view taken by the TCSEC is that of an "atomic" TCB; the TCB is seen to be a single entity which is, in some sense, homogeneous. This allows a relatively simple measure (i.e., the digraphs) to be assigned to the TCB. However, real systems may be more complex, resulting in the inability of a single, simple rating to convey the full complexity of the system. This is implicitly recognized in TCSEC evaluation reports and EPL entries, in which credit may be given to a vendor for meeting TCSEC (functional) requirements beyond those necessary to satisfy the rating (e.g., the B3 discretionary access control feature in a C2 TCB). In short, systems which reflect straightforward implementations or extensions of the TCSEC can accurately be described with a single digraph. On the other hand, adding complexity to systems may violate assumptions which underlie the TCSEC rating system, requiring a more complex description if accuracy is to be achieved.

If a TCB made up of TCB subsets is consistent with the TCSEC assumptions on homogeneity, then a simple digraph suffices for a full and accurate description of the security properties of the product. However, to the extent that a subsetted architecture introduces complexity not captured by the digraphs, the simple TCSEC ratings cannot be applied to the composite system. More specifically, for a subsetted TCB to achieve a single rating, all the requirements of that class must be satisfied. For example, if a discretionary access control-enforcing DBMS TCB subset is added onto a previously evaluated B3 product, the entire system can achieve a B3 rating if it could also have achieved the B3 rating evaluated as a monolith. That is, the new TCB subset must also satisfy all the assurance and architectural requirements of B3.

Consider a candidate TCB subset which enforces a discretionary access control policy over a new type of object, targeted at a host system which has already been evaluated at the B3 level. Examples are a database management system providing discretionary access control over tuples, a transaction processor providing discretionary access control over transactions, and a message system providing discretionary access control over messages. If the candidate TCB subset meets all the C2 requirements, the problem is what rating will be assigned to the composite system. To designate it a "C2" is clearly inaccurate, as well as being unfair to the original B3 product vendor. To designate it "B3" may be equally inaccurate, and it creates ambiguity in the meaning of the metric used for comparing systems. In fact, depending on the details of the specific candidate, the composite system could legitimately be rated at any level from C2 to B3 under a TCSEC evaluation.

The TCSEC rating system assumes a measure of homogeneity which the above example violates thus invalidating the very basis upon which a single digraph may be assigned. Hence, a subsetted system such as described above, will have to be characterized with a more complex description than a single digraph. Although this may seem undesirable, it will be a more accurate description of the system, and it provides sufficient information to allow system designers and accreditors to make decisions about sufficiency of security for their specific applications. In essence, such an approach is necessary for recognizing the additional complexity which can be introduced by architectures which allow system elements to be developed separately.

GLOSSARY

candidate TCB subset The identification of the hardware, firmware, and software that make up the proposed TCB subset, along with the identification of its subjects and objects; one of the conditions for evaluation by parts.

content-dependent access control Access control in which access is determined by the value of the data to be accessed.

context-dependent access control Access control in which access is determined by the specific circumstances under which the data is being accessed.

database management system A computer system whose main function is to facilitate the sharing of a common set of data among many different users. It may or may not maintain semantic relationships among the data items.

DBMS Abbreviation for "database management system."

depends A TCB subset A depends (for its correctness) on TCB subset B if and only if the (engineering) arguments of the correct implementation of A with respect to its specification assume, wholly or in part, that the specification of B has been implemented correctly.

domain The set of objects that a subject has the ability to access.

dominated by Security level A is dominated by security level B if (1) the clearance/classification in A is less than or equal to the clearance/classification in B, and (2) the set of access approvals (e.g., compartment designators) in A is contained in the set of access approvals in B (i.e., each access approval appearing in A also appears in B). This dominance relation is a special case of a partial order.

dominates "Security level B dominates security level A" is synonymous with "security level A is dominated by security level B." See "dominated by."

global requirements Those which require analysis of the entire system and for which separate analysis of the individual TCB subsets does not suffice. See Section TC-5.3.2 for a summary list.

lattice A partially ordered set for which every pair of elements has a greatest lower bound and a least upper bound.

local requirements Those for which separate analysis of the individual TCB subsets suffices to determine compliance for the composite TCB. See Section TC-5.3.1 for summary list.

metadata (1) Data referring to other data; data (such as data structures, indices, and pointers) that are used to instantiate an abstraction (such as "process," "task," "segment," "file," or "pipe"). (2) A special database, also referred to as a data dictionary, containing descriptions of the elements (e.g., relations, domains, entities, or relationships) of a database.

monolithic TCB A TCB that consists of a single TCB subset.

object A passive entity that contains or receives information. Access to an object potentially implies access to the information it contains. Examples of objects are: records, blocks, pages, segments, files, directories, directory trees, and programs, as well as bits, bytes, words, fields, processors, video displays, keyboards, clocks, printers, network nodes, etc.

partial order A relation that is symmetric (a is related to a), transitive (if a is related to b and b is related to c, then a is related to c), and antisymmetric (if a is related to b and b is related to a, then a and b are identical.)

primitive An ordering relation between TCB subsets based on dependency (see "depends" above). A TCB subset B is more primitive than a second TCB subset A (and A is less primitive than B) if (a) A directly depends on B or (b) a chain of TCB subsets from A to B exists such that each element of the chain directly depends on its successor in the chain.

reference monitor concept An access control concept that refers to an abstract machine that mediates all accesses to objects by subjects.

reference validation mechanism "An implementation of the reference monitor concept . . . that validates each reference to data or programs by any user (program) against a list of authorized types of reference for that user." It must be tamper proof, must always be invoked, and must be small enough to be subject to analysis and tests, the completeness of which can be assured. [1]

security policy The set of laws, rules, and practices that regulate how an organization manages, protects, and distributes sensitive information.

storage object An object that supports both read and write accesses.

subject An active entity, generally in the form of a person, process, or device that causes information to flow among objects or changes the system state. Technically, a process/domain pair.

subset-domain A set of system domains. For evaluation by parts, each candidate TCB subset must occupy a distinct subset domain such that modify-access to a domain within a TCB subset's subset-domain is permitted only to that TCB subset and (possibly) to more primitive TCB subsets.

TCB subset A set of software, firmware, and hardware (where any of these three could be absent) that mediates the access of a set S of subjects to a set O of objects on the basis of a stated access control policy P and satisfies the properties:

(1) M mediates every access to objects O by subjects in S;

(2) M is tamper resistant; and

(3) M is small enough to be subject to analysis and tests, the completeness of which can be assured.

technical policy The set of rules regulating access of subjects to objects enforced by a computer system.

Trusted Computing Base (TCB) The totality of protection mechanisms within a computer system including hardware, firmware, and software the combination of which is responsible for enforcing a security policy. A TCB consists of one or more components that together enforce a unified security policy over a product or system. The ability of a TCB to correctly enforce a security policy depends solely on the mechanisms within the TCB and on the correct input by system administrative personnel of parameters (e.g., a user's clearance) related to the security policy.

trusted subject A subject that is permitted to have simultaneous view- and alter-access to objects of more than one sensitivity level.

user Any person who interacts directly with a computer system.

view That portion of the database that satisfies the conditions specified in a query.

view definition A stored query; sometimes loosely referred to as a "view."

BIBLIOGRAPHY

1. J. P. Anderson, "Computer Security Technology Planning Study," ESD-TR-73-51 (AD-758206), J. P. Anderson Co., October 1972.

2. J. P. Anderson, "On the Feasibility of Connecting RECON to an External Network," Technical Report, J. P. Anderson Co., March 1981.

3. D. E. Bell and L. J. La Padula, "Secure Computer Systems: Unified Exposition and Multics Interpretation," MTR-2997, (AY/W 020 445), The MITRE Corporation, Bedford, Massachusetts, July 1975.

4. D. E. Bell and L. J. La Padula, "Secure Computer Systems: Mathematical Foundations," MTR-2547-I, (AD 770 768), The MITRE Corporation, Bedford, Massachusetts, March 1973.

5. L. J. La Padula and D. E. Bell, "Secure Computer Systems: A Mathematical Model," MTR 2547-II, (AD 771 543), The MITRE Corporation, Bedford, Massachusetts, May 1973.

6. D. E. Bell, "Secure Computer Systems: A Refinement of the Mathematical Model," MTR 2547-III, (AD 780 528), The MITRE Corporation, Bedford, Massachusetts, December 1973.

7. D. E. Bell, "Secure Computer Systems: A Network Interpretation," Proceedings of the Second Aerospace Computer Security Conference, McLean Virginia, December 2-4, 1986, pp. 32-39.

8. Department of Defense, Department of Defense Trusted Computer System Evaluation Criteria, DOD 5200.28-STD, December 1985.

9. D. E. Denning, "Cryptographic Checksums for Multilevel Database Security," Proceedings of the IEEE Symposium on Security & Privacy, Oakland, California, April 29-May 2, 1984, pp. 52-61.

10. D. E. Denning, "Commutative Filters for Reducing Inference Threats in Multilevel Database Systems," Proceedings of the IEEE Symposium on Security & Privacy, Oakland, California, April 22-24, 1985, pp. 134-146.

11. E. B. Fernandez, R. C. Summers, and C. Wood, Database Security and Integrity, Boston, Massachusetts:

Addison Wesley, 1981.

12. C. Garvey and A. Wu, "ASD Views," Proceedings of the Fourth Aerospace Computer Security Applications Conference, Orlando, Florida, December 1988, pp. 85-95.

13. R. D. Graubart and J. P. L. Woodward, "A Preliminary Naval Surveillance DBMS Security Model," Proceedings of the IEEE Symposium on Security & Privacy, Oakland, California, April 26-28, 1982, pp. 21-37.

14. R. D. Graubart, "The Integrity-Lock Approach to Secure Database Management," Proceedings of the IEEE Symposium on Security & Privacy, Oakland, California, April 29-May 2, 1984, pp. 62-74.

15. M. J. Grohn, "A Model of a Protected Data Management System," ESD-TR-76-289, I. P. Sharp Associates, Ltd., June 1976.

16. T. H. Hinke, M. Schaefer et al., "Secure Data Management System," RADC-TR-75-266 (AD-A019201), System Development Corporation, Santa Monica, California, November 1975.

17. C. E. Landwehr, C. L. Heitmeyer, and J. McLean, "A Security Model for Military Message Systems," ACM Transactions on Computer Systems, Vol. 2, No. 3, August 1984, pp. 198-222.

18. T. F. Lunt, D. E. Denning, P. G. Neumann, R. R. Schell, M. Heckman, and W. R. Shockley, "Final Report Vol. 1: Security Policy and Policy Interpretation for a Class A1 Multilevel Secure Relational Database System," Computer Science Laboratory, SRI International, Menlo Park, California, 1988.

19. J. McHugh and B. M. Thuraisingham, "Multilevel Security Issues in Distributed Database Management Systems," Computers & Security, Vol. 7, No. 4, Elsevier Advanced Technology Publications, August 1988, pp. 387-396.

20. National Computer Security Center, Proceedings of the National Computer Security Center Invitational Workshop on Database Security, Baltimore, Maryland, June 17-20, 1986.

21. P. A. Rougeau and E. D. Sturms, "The Sybase Secure Dataserver: A Solution to the Multilevel Secure DBMS Problem," Proceedings of the 10th National

Computer Security Conference, Baltimore, Maryland, September 21-24, 1987, pp. 211-215.

22. M. Schaefer, ed., Multilevel Data Management Security, Air Force Studies Board, Committee on Multilevel Data Management Security, National Academy Press: Washington, D.C., 1983.

23. M. D. Schroeder and J. H. Saltzer, "A Hardware Architecture for Implementing Protection Rings," Communications of the ACM, Vol. 15, No. 3, March 1972, pp.157-170.

24. W. R. Shockley and R. R. Schell, "TCB Subsets for Incremental Evaluation," Proceedings of the Third Aerospace Computer Security Conference, Orlando, Florida, December 7-11, 1987, pp. 131-139.

25. P. Stachour and B. Thuraisingham, "Design of LDV - A Multilevel Secure Database Management System," IEEE Transactions on Knowledge and Data Engineering, Vol. 2, No. 2, June 1990, pp. 190-209.

26. M. Stonebraker, "Operating System Support for Database Management," Communications of the ACM, Vol. 24, No. 7, July 1981, pp. 412-418.

27. Unisys Corporation, "Secure Distributed Database Management System," Final Technical Report, Rome Air Development Center Technical Report, RADC-TR-89-314, Vol. 1-5, December 1989.

28. J. Wilson, "Views as the Security Objects in a Multi-level Secure Relational Database Management System," Proceedings of the 1988 IEEE Symposium on Security & Privacy, Oakland, California, April 18-21, 1988, pp. 70-84.