GUIDE TO UNDERSTANDING TRUSTED FACILITY MANAGEMENT

June 1989

NATIONAL COMPUTER SECURITY CENTER
NCSC-TG-O15
Library No. S-231, 429

FOREWORD

The National Computer Security Center (NCSC) has established an aggresive program to study and implement computer security technology and to encourage the widespread availability to trusted computer operations. To provide insight into the Trusted Computer Systems Evaluation Criteria (TCSEC) and to assure that each feature of the TCSEC will be discussed in detail and provide the proper interpretation with specific guidance, the NCSC has established a Technical Guideline Program This Technical Guideline Program, and the cooperative business relationship being forged with the computer and telecommunication industries, will result in the fulfillment of our country's computer security requirement. We are determined to meet the challenge of identifying trusted computer guidelines suitable for use in processing all types and classifications of information.

"A Guide to Understanding Trusted Facility Management" is the latest in the series of technical guidelines that are being published by the National Computer Security Center. This technical guideline has been written to help the computer security manufacturers, system evaluators, accreditors, as well as end users understand what procedures, methods, and processes are required for trusted facility management at B2 through A1 classes ofthe TCSEC.

As the Director, National Computer Security Center, I invite your recommendations for revision to this technical guideline. We plan to review this document periodically or when the need arises.

Patrick R. Gallagher Jr. 15 August 1989
Director
National Computer Security Center

ACKNOWLEDGEMENTS

Special recognition for their contributions to this document are extended to Info Systems Technology, Inc., and to Dr. Virgil D. Gligor of the University of Maryland as primary author of this document, and to Ms. Valerie A. Maurer and MAJ James P. Gordon (U S Army) as Project Managers for the production and preparation of this guideline.

Acknowledgment is given to the many computer vendor representatives, and members of the National Computer Security Center (NCSC) community, who enthusiastically gave of their time and technical expertise in reviewing the guideline and providing valuable comments and suggestions. Special thanks is given to Ms. Carol Lane, Mr. Leon Howell and Mr. Douglas Hardie for their invaluable assistance and guidance in this effort.

PREFACE

This guideline contains information derived from the requirements of the TCSEC prefaced by the word "shall", and recommendations derived from good practices prefaced by the word "should" when conducting trusted facility management. The recommendations in this document are also not to be construed as supplementary requirements to the TCSEC. The TCSEC is the only metric against which systems are to be evaluated.

Throughout this guideline there will be examples, illustrations, or citations of administrative roles and operations that have been used in trusted facility management. The use of these examples, illustrations, and citations does not mean that they contain the only acceptable procedures, methods, or processes. The selection of these examples is based solely on their availability in the computer security literature. Examples in this document are not to be construed as the only implementations that will satisfy the TCSEC requirements or intended to single out any particular operating system to highlight weaknesses and shortfalls, but merely to provide clarification. The examples are suggestions of appropriate implementations.

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

The principal goal of the National Computer Security Center is to encourage the widespread availability of trusted computer systems. In support of that goal a metric was created, the DoD Trusted Computer System Evaluation Criteria (TCSEC), against which computer systems could be evaluated for security. The TCSEC was originally published on 15 August 1983 as CSC-STD-001-83. In December 1985 the DoD adopted it, with a few changes, as a DoD Standard, DoD 5200.28-STD. DoD Directive 5200.28, "Security Requirements for Automated Information Systems (AISs)", has been written, among other reasons, to require the Department of Defense Trusted Computer System Evaluation Criteria be used throughout the DoD. The TCSEC is the standard used for evaluating the effectiveness of security controls built into AISs. The TCSEC is divided into four divisions: D, C, B, and A, ordered in a hierarchical manner with the highest division (A) being reserved for systems providing the best available level of assurance. Within divisions C , B, and A, there are subdivisions known as classes, which are also ordered in a hierarchical manner to represent different levels of security.

1.1. PURPOSE

An important assurance requirement of the TCSEC, which appears in all classes from B2 to A1, is trusted facility management. This refers to the administrative procedures, roles, functions (e.g., commands, programs, interfaces), privileges and databases that are used for secure system configuration, administration and operation.

The objective of trusted facility management is to support security and accountability policies throughout a system's operation. To accomplish this goal, two key requirements are the separation between Administrator and Operator functions, in class B2, and between security-relevant and nonsecurity-relevant functions of System Administrators, in class B3. This separation of administrative and operator functions, and security-relevant and nonsecurity-relevant functions of System Administrators, also applies to class A1. These separations help ensure that security-adverse effects of human error, misdeed, and system failure do not affect administrative functions and data.

The purpose of "A Guide to Understanding Trusted Facility Management" is to provide guidance to manufacturers on how to incorporate functions of trusted facility management into their systems; to system evaluators and accreditors on how to evaluate the design and implementation of trusted facility management functions; and to end users on how to use these functions effectively, e.g., on how to avoid common pitfalls of system management.

1.2. SCOPE

The guidelines for trusted facility management presented herein refer to the separation of administrative functions, interfaces, and procedures of an important assurance requirement of classes B2 through A1 of the TCSEC. This guideline is intended to present the issues involved in the design of trusted facility management.

This guideline contains five.additional sections. Section 2 contains a brief overview of the inherent vulnerabilities of administrative roles. Section 3 presents TCSEC requirements that affect the design and implementation of trusted facility management functions, and includes recommendations corresponding to each evaluation class. Section 4 reviews the major requirements of trusted facility management as stated in the TCSEC. Section 5 presents the separation between Administrator's and Operator's functions and the possible partitioning of the security-relevant functions of the Administrator and Operator into separate roles, functions and databases. Section 6 discusses the impact of the other TCSEC requirements on trusted facility management, including design and modeling alternatives for trusted facility management.

Not addressed herein are personnel security measures, physical security of the automated information system equipment, and other administrative measures external to the AIS. The evaluation of these measures is beyond the scope of TCSEC-based evaluations [12, p.87]. These guidelines apply to computer systems, processing environments, and products built or modified with the intention of satisfying the TCSEC requirements. Note that this document contains suggestions and recommendations derived from TCSEC objectives but which are not required by the TCSEC. Additional recommendations are made, which are derived from the stated objectives of the TCSEC.

1.3. CONTROL OBJECTIVES

Trusted facility management is one of the areas of operational assurance. As such, the trusted facility management is an aspect of the objective, "assurance." The assurance objective provided in the TCSEC is:

"Systems that are used to process classified or other sensitive information must be designed to guarantee correct and accurate interpretation of the security policy and must not distort the intent of that policy. Assurance must be provided that correct implementation and operation of the policy exists throughout the system's life cycle."

This objective affects trusted facility management in two important ways. First, administrative roles of the system are the key components that help to ensure the enforcement of the system security policy, and thus, their function must support the intent of that policy. Second, the administrative roles must satisfy the life-cycle assurance requirements of correct implementation and operation.

2. SECURITY ADMINISTRATION - THE PROBLEM

Weaknesses of trusted facility management are role specific and common to all administrative roles. Careful examination of both common administrative roles and role-specific weaknesses is important for both system designers and administrators because exposure to some of these weaknesses can be reduced or eliminated by specific designs or by administrative procedure external to the system in use. The distinction between the two types of weaknesses is also useful for the strengthening of mechanisms and procedures supporting different roles selectively.

The weaknesses discussed below are generic in the sense that they are not specific to any particular system or design. Careful analysis should be performed in designing and implementing specific systems to identify specific additional weaknesses and their required countermeasures. Design, implementation, and use of auto*mated tools for analyzing specific system weaknesses are useful, but still a research subject [1].

Three types of weaknesses affect all administrative roles to various degrees:

(1) unauthorized modification of hardware and software system configuration Unauthorized changes of system configuration, including both hardware and software changes, can take place during all phases of a system life-cycle.

(2) penetration of a specific administrative role. Penetration of administrative roles by non-administrative users, or by unauthorized administrative users, is usually made possible by flawed, or weak, mechanisms for identification and authentication, TCB protection, or role separation.

(3) misuse of administrative authority. This can arise from careless or deliberate misuse of administrative authority. Misuse of authority can cause both TCB and user security violations, and therefore can lead to extensive damage.

3. TCSEC REQUIREMENTS FOR TRUSTED FACILITY MANAGEMENT

In the TCSEC, n requirements for Trusted Facility Management are for security classes B2 through A1. Classes C1 through B1 have no Trusted Facility Management requirements.

3.1. REQUIREMENTS FOR SECURITY CLASS B2

3.1.1. Security Policy

No Additional Requirements.

3.1.2. Accountability

All identification and authentication requirements of class B2, including trusted path, shall apply to the administrative users individually.

All actions of administrative users shall be auditable in accordance with the B2 audit requirements.

3.1.3. Operational Assurance

3.1.3.1. System Architecture

The TCB programs and data structures implementing administrative functions:

* must satisfy the modularity requirements of class B2;

* must satisfy the least privilege principle;

* must use logically distinct storage objects with separate attributes (e.g., files, segments).

The interfaces of the administrative roles implemented by the TCB must be completely defined, and all the elements of the TCB implementing the administrative roles must be identified.

3.1.3.2. Trusted Facility Management

The TCB shall support separate Operator and Administrator functions. The Administrator's functions include those of:

* the Security Administrator

* System Programmer

* the Auditor

* the Account Administrator
(whenever this role is defined to be security-relevant).

These functions must be separated from those of the Secure Operator. While the Administrator's functions may be combined into one function, we recommend they be separated as described in section 5. The remaining functions include only the nonsecurity-relevant functions.

3.1.4. Life-Cycle Assurance

3.1.4.1. Security Testing

All security testing requirements of class B2 apply to the TCB functions and interfaces implementing administrative roles as stated.

3.1.4.2. Design Specification and Verification

Recommendation:

-Descriptive Top-Level Specifications (DTLSs) of the TCB functions and interfaces implementing administrative roles must be maintained that completely and accurately describe these functions and interfaces in terms of exceptions, error messages, and effects.

-A formal security and integrity model of trusted facility management should be used to define the separation of administrative roles, functions, privileges and databases.

3.1.4.3. Configuration Management

All configuration management requirements of class B2 apply to the

TCB functions and interfaces implementing administrative roles as stated.

3.1.5. Documentation

3.1.5.1. Trusted Facility Manual

A manual shall be available that provides the following:

* be addressed to the ADP system administrator shall present cautions about functions and privileges that should be controlled when running a secure facility.

* give procedures examining and maintaining the audit files.

* give the detailed audit record structure for each type of audit event.

* describe the operator and administrator functions related to security, to include changing the security characteristics of a user.

* provide guidelines on the consistent and effective use of the protection features of the system.

* explain how the protection features of the system interact.

* show how to securely generate a new TCB.

* provide guidelines on facility procedures, warinings, and privileges that need to be controlled in order to operate the facility in a secure manner.

* identify the TCB modules that contain the reference validation mechanism.

* describe the procedures for secure generation of a new TCB from source after modification of any modules in the TCB.

3.1.5.2. Test Documentation

All test documentation requirements of class B2, except those for covert channel testing, apply to the TCB functions and interfaces implementing administrative roles as stated.

3.1.5.3. Design Documentation

Documentation shall be available that provides a description of:

* Interfaces between the TCB modules implementing functions of the administrative roles;

* Specific TCB protection mechanisms used for the separation of administrative roles;

* Descriptions of the TCB modules implementing functions and

interfaces of the administrative roles;

* How the least privilege principle is supported by the functions and interfaces of the TCB implementing administrative roles;

* How the actions of the administrative roles are audited.

Recommendation:

-A formal description of the security and integrity policy model used to define the separation of administrative roles should be available and proven to be sufficient to enforce the claimed separation.

3.2. REQUIREMENTS FOR SECURITY CLASS B3

All the requirements of Class B2 are included at this level. The additional class B3 requirements are listed below.

3.2.1. Security Policy

No Additional Requirements.

3.2.2. Accountability

The trusted-path requirements of class B3 apply to administrative users.

The additional audit requirements of class B3 apply to the administrative users.

3.2.3. Operational Assurance

3.2.3.1. System Architecture

The additional TCB structuring requirements of class B3 (i.e., significant use of abstraction, information hiding, and layering) apply to the functions and interfaces of the TCB implementing administrative roles.

3.2.3.2. Trusted Facility Management

The security-relevant administrative functions (i.e., those of the Security Administrator, System Programmer, Auditor and the Secure Operator's roles defined above) must be separated from the nonsecurity-relevant administrative functions.

The security-relevant administrative functions must be limited to those that are essential to performing the security roles effectively.

All actions of security personnel (Secure Administrator and Secure Operator) must be audited.

Recommendations:

- The functions of security administration and personnel should distinguish among

* System Programmer, Security Administrator, Auditor, and Secure Operator

* their privileges

* their databases.

- Different levels of trust should be established for the following roles in accordance with the power and vulnerability of each role:

* System Programmer (maintenance and diagnostics mode);

* Security Administrator;

* Auditor;

* Secure Operator;

* Account Administrator;

* Operator.

(Note: The distinction between the System Administrators, Operators, and System Security Officers is explicitly made in the audit requirements of the TCSEC [11, p. 16]. These roles correspond to the Account Administrator, Secure/Normal Operator, and Security Administrator/ Auditor roles above. Also note that these distinctions do not require the separation of security-relevant and nonsecurity-relevant functions as they are made in the audit -- not trusted facility management -- requirement area).

3.2.3.3. Trusted Recovery

The trusted recovery requirement of class B3 applies to the functions and interfaces of the TCB implementing administrative roles.

3.2.4. Life-Cycle Assurance

3.2.4.1. Security Testing

All additional security testing requirements of class B3 apply to the functions and interfaces of the TCB implementing administrative roles.

3.2.4.2. Design Specification and Verification

Recommendation:

- The additional design specification and verification requirements of class B3 should be applied to the functions and interfaces of the TCB implementing administrative roles.

3.2.4.3. Configuration Management

No Additional Requirements.

3.2.5. Documentation

3.2.5.1. Trusted Facility Manual

The additional requirements shall include procedures to ensure that the system is initially started in a secure state and procedures to resume secure system operation after any lapse in system operation.

3.2.5.2. Test Documentation

No Additional Requirements.

3.2.5.3. Design Docu*menta*tion

No Additional Requirements.

3.3. REQUIREMENTS OF SECURITY CLASS A1

All requirements of the security class B3 are included here. The only additional requirements are in the following "Life-Cycle Assurance" areas:

3.3.1. Additional Life-Cycle Assurance Requirements

3.3.1.1. Configuration Management

All additional configuration management requirements of class A1 apply to the TCB functions and interfaces implementing administrative roles.

3.3.1.2. Trusted Distribu*tion

All trusted distribution requirements of classA1 apply to the TCB functions and interfaces implementing administrative roles.

4. SATISFYING THE TCSEC REQUIREMENTS

The principal requirements of trusted facility management are:

* the separation of Operator and Administrator functions;

* the logical (or physical) separation of the database information corresponding to those functions; and

* the implementation of least privilege such that functions have only the minimum necessary privileges to the databases.

4.1. SEPARATION OF ADMINISTRATOR AND OPERATOR FUNCTIONS

The separation of Administrator and Operator functions is a requirement of TCSEC class B2, which states:

"The TCB shall support separate Operator and Administrator functions."

The primary purpose behind the separation of the Operator and Administrator functions is to limit the potential damage that untrusted, or errant, code can inflict on the information the TCB uses to enforce the security policy. Any code executed with Operator or Administrator privileges has the ability to change the TCB data structures, thus affecting the enforcement of policy. Through the application of the principal of least privilege and the separation of Operator and Administrator functions so that they are prevented from executing untrusted code, the TCB data structures can be protected. The principle of least privilege requires that each subject be granted the most restrictive set of privileges needed for the specific task. In the case of the operator and administrator functions, the privileges need to be established at a low level of granularity so that the proceses that implement those functions do not have unnecessary privileges. This low level of granularity provides several important protections:

* limits the effects of errors on the part of the administrator;

* limits the effects of incorrect code which implements the administrator functions;

* provides some protection against malicious administrators, in that damage that can be done is strictly contained to the provileges defined for that role. Some additional protection is afforded by the auditing of administrator actions. (This argument can be extended to malicious code which is inserted in the administrator functions.)

The TCSEC recognizes the need to separate the operator and adminstrator functions from the normal user abilities to execute code. There are several ways to implement such separation. One way is to enforce those restrictions on the Administrator and Operator functions. They can only execute trusted code that has been shown to preserve the TCB data structures properly. This requires that the people who perform those functions also have a separate account that allows them to be a normal user. That separate account would not have any Operator or Administrator capabilities. Whatever approach to separation is selected, it must be shown to restrict the Operator or Administrators from executing untrusted code.

The separation of Operator and Administrator functions, namely between the commands, programs, and interfaces implementing those functions, is important because these functions are used with different privileges, on different system data. Should these functions not be separated, Operators could use commands that include Administrators' privileges and databases. This would mean that all Operators would need to be trusted to the same degree as that needed for Administrators. It would also mean that the principles of least privilege and separation of privilege, which are two of the most important security principles (see reference [18] for a further explanation of these principles), are violated, overexposing the system to error, failure, and misdeed. Furthermore, lack of functional separation would fail to confine the effects of any function penetration, leaving the entire system in a vulnerable state.

In addition to the separation of Administrator and Operator functions, trusted facility management should also separate internal system databases which the Operator and Administrator manipulate. Checks and balances are necessary to avoid trusting too many all-powerful Administrators. The identification of the security-relevant, internal system databases and the correlation between each function and the corresponding database shall be carefully performed and documented. The separation of Operator's and Administrator's functions shall also lead to the separation of accessible objects and of access privileges to shared databases. This is an essential design requirement for the enforcement of the least privilege principle within the TCB because it helps identify and eliminate unnecessary Operator access to administrator data. For example, the Administrator has full access to system databases that need not be fully accessible to the Operator; i.e., the Administrator has Read/Write privileges to some (shared) databases, such as the system security profile, for which the Operator only needs Read privileges. Thus, the Write privilege of the Operator to these databases would be eliminated. Also, because these databases are separate, consistency checks may be derived from the security-relevant databases of the Administrator and applied to the security-relevant functions of the Operator. This would increase the robustness of the administrative functions of the system and, implicitly, its usefulness.

Figure 1 illustrates both the separation of function and of privileges/databases for class B2. Note, although the functions of the Operator and Administrator are completely separated, the Administrator's privileges include those of the Operator in the sense that the Administrator can always get access to all Operator functions, databases, and privileges. For example, an Administrator can always log in as an Operator and perform Operator functions. In contrast, the Operator cannot get access to functions, databases, and privileges that are exclusively the Administrator's. Note, this hierarchical relationship of roles is a functional hierarchy. The system could provide a "flat" set of roles, functions and privileges, and the hierarchy could be managed administratively.

4.1.1. Security-Relevant Functions of the System Administrator

The security-relevant functions of the System Administrator include those that:

* Define and change the user security characteristics and those of the system security data (e.g., user identifier, user's group identifiers, user/group maximum security level; and the maximum/minimum security level of the system data, the maximum/minimum security level of each file system).

* Define and change the system's security characteristics (e.g., security level limits of multilevel channels, I/O processors, communication lines, and devices; all possible level changes of single level devices).

* Perform system programming functions; (e.g., trusted system configuration in accordance with the configuration management policy, system distribution, system installation, TCB code maintenance that may affect system configuration, distribution and installation).

* Perform audit functions (e.g., determine what events should be audited, manage the audit trail, analyze the audit trail, produce audit reports).

4.1.2. Security-Relevant Functions of the Operator

The security-relevant functions of the Operator include those that:

* Enable and disable peripheral devices, make changes to the device security characteristics within the limits defined by the Administrator (e.g., the Operator sets the level of a single-level device within the range defined by the Administrator).

* Control the mounting of file systems and load labeled disk packs and tape reels on appropriate drives.

* Recover user files following system crashes.

* Handle printed output.

* Perform maintenance operations on user databases and routine maintenance of TCB databases.

* Boot up and shut down the system.

4.2. SEPARATION OF SECURITY AND NONSECURITY-RELEVANT FUNCTIONS

The second requirement of the trusted facility management is to identify, audit, and separate the security-relevant functions of the Administrator from the nonsecurity-relevant functions. The purpose of this requirement is to prevent an Operator or Administrator from executing untrusted code using their special privileges that would enable that code to corrupt the policy enforcement data or mechanisms. This requirement is introduced in class B3, and is stated in the TCSEC as follows:

"The functions performed in the role of a Security Administrator shall be identified. The AIS administrative personnel shall only be able to perform Security Administrator functions after taking a distinct auditable action to assume the Security Administrator role on the AIS. Nonsecurity functions that can be performed in the Security Administrator role shall be limited strictly to those essential to performing the security role effectively."

Both the Administrator and the Operator roles include security-relevant functions. Security-relevant functions include all administrative functions that are used to implement the security and accountability policies supported by a system. Nonsecurity-relevant functions are those that cannot affect the implementation of security and accountability policies supported by a system. The separation of security-relevant and nonsecurity-relevant functions is important because nonsecurity-relevant functions need to be trusted to a degree lower than that of the security-relevant ones. A higher degree of trust implies that the operational and life-cycle assurance tasks are more extensive than those necessary for functions of a lower level of trust. Although some nonsecurity-relevant functions of the Administrator may be functionally a part of the TCB in class B2, flaws in these functions should lead only to potential denial-of-service instances, but not to security or integrity violations. In class B3, essentially where the nonsecurity-relevant functions of the Administrator shall be removed from the TCB. The TCSEC does permit the inclusion of nonsecurity relevant functions that are essential to performing the security role. While the separation of administrative functions is not required below class B2, the benefits and protection it provides should be seriously considered.

Figure 2 illustrates both the separation of function and of privileges/databases for classes B2 and B3. Note, although the functions of the Operator and Security Administrator (i.e., the nonsecurity-relevant role of the Administrator) are completely separated.

(Alternative administrative procedures for systems that do not support any separation of roles have been suggested [5]. These procedures may be useful for systems in TCSEC classes C1 through B1.)

4.3. IMPACT OF OTHER TCSEC REQUIREMENTS ON TRUSTED FACILITY MANAGEMENT

The third important requirement of trusted facility management is the integration of functions and programs that implement administrative roles within the TCB in such a way that the security policy, accountability, assurance, and documentation requirements of specific TCSEC classes are satisfied. For example, in a B3 or above system, the design of each function supporting a specific role must ensure that the programs executing that function operate with the fewest privileges necessary and that they are designed to satisfy the abstraction, information hiding, and layering requirements. Furthermore, in a class B3 or above system, the nonsecurity-relevant functions of Administrators shall be removed from the TCB because "significant system engineering shall be directed towards minimizing the complexity of the TCB and excluding from the TCB modules that are not protection critical" [11]. Some work environments require the system to support multiple work shifts. Such a system design, allowing multiple individuals to belong to the same role, shall ensure that these individuals are not forced to share a role password, such that accountability on an individual basis is lost.

Most documentation requirements of the TCSEC apply to trusted facility management as stated in each evaluation class. However, some requirements such as those that state the need for a Security Features Users' Guide (SFUG) and for covert channel analysis are obviously not applicable. The SFUG is relevant for all users, whereas the Trusted Facility Manual and Management are relevant only for administrative users. Also, since most administrative users have multilevel access to system and user data, they must be trusted to maintain the secrecy and classification of the data. Thus, administrative users must be cleared to the highest level of data classification. Furthermore, all code implementing functions of administrative roles should be scrutinized to ensure, to the largest extent possible, that it does not contain any Trojan horses or trap doors. Additional requirements imposed by the TCSEC of trusted facility management are discussed in the section entitled, "TCSEC Requirements For Trusted Facility Management."

5. SEPARATION OF OPERATOR'S AND ADMINISTRATOR'S ROLES

An important aspect of trusted facility management is that of partitioning the security-relevant duties of the Administrators and Operators into separate roles. For example, this partitioning could distinguish the security-relevant roles of Security Administrator, System Programmer, and Auditor -- in addition to the non-security-relevant role of Accounts Administrator; and also could distinguish between the security-relevant functions of the Operator (the Secure Operator role) and the nonsecurity-relevant ones (the Operator role). Although this further partitioning of the Administrator's duties is not required by the TCSEC, it is suggested:

(1) by the need to differentiate between the skills required by different security-relevant functions of the Administrator and Operator,

(2) by the need to divide the power (e.g., privileges) of the all-encompassing Administrator duty into multiple roles that incorporate different levels of trust,

(3) by the need to avoid entrusting all security-relevant functions to a single role or individual. In this partitioning of the Administrator's duties, the Security Administrator role retains the functions of defining and changing the users' and the system security profiles.

The System Programmer's functions differ from those of the Security Administrator, Auditor, Account Administrator and Operators. The System Programmer's functions, privileges, and databases include those of the other roles, as the System Programmer is the most privileged administrative user defined in any system. In contrast with the other roles, some of the System Programmer's actions may not be auditable. This is the case because some of the System Programmer's actions take place before the Auditor's programs and databases are configured and loaded. Furthermore, the System Programmer's maintenance activities may refer to the maintenance/repair of the TCB, including the other roles' interfaces (e.g., commands, programs), databases, and privileges. Whenever possible, the System Programmer functions should be relegated to system maintenance mode only and monitored by administrative procedure. Whenever possible, work on TCB code should be done on a developmental system rather than on a system in current use. The developmental system may be a physically separate system or a system from which user data, and in particular classified data, have been removed (e.g., by changing disk packs or overwriting memory) prior to performing TCB maintenance. Note that any modification of the TCB code, even by authorized users in the System Programmer role, may invalidate the system's rating. The above measures allow the design of a system whose mode of operation does not include an all-powerful role.

The Auditor's functions, databases, and access privileges differ significantly from those of the other administrative roles (e.g., Security

Administrator, Account Administrator, Operators). The separation of the Auditor's functions, databases, and access privileges from those of the Security Administrator, Account Administrator, and Operators is an important application of the separation of privilege and least privilege principles. Should such separation not be performed, and should the Security Administrator be allowed to undertake Auditor functions or vice-versa, the entire security function would become the responsibility of a single, unaccountable individual or role in normal mode of system operation. For example, a Security Administrator may take actions that represent misuse of authority and then use Auditor functions to erase any evidence of his actions. Although this is obviously undesirable, the TCSEC does not require the separation of Security Administrator and Auditor functions (and neither does it require the separation between Secure Operator and Operator functions).

Figure 3 illustrates both the fine-grained separation of roles and of databases/privileges. The relationships between the different roles defined here are explained in Section 5.8.

The design of each administrative role should include explicitly the set of commands, privileges, and databases specific to that role. In contrast, the assignment of individuals to the roles is best left to the management of the installations familiar with the skill, interests, and trust that can be assigned to the individuals. Furthermore, this guide does not distinguish between the role of the System Programmer of a specific installation and that assigned to a manufacturer's programmer. Such distinctions depend on the operational environment and administrative procedures enforced in that environment. In small system environments the two roles become indistinguishable, whereas in large system environments the two roles are different. In some environments, the System Programmer has the right to examine, modify, recompile, and rebuild the TCB, whereas in others the System Programmer can only install a given object code version of it. For example, it is not uncommon that System Programmers at a given installation site add device drivers to a TCB for new multilevel devices supported in the systems, and then rebuild the TCB. Whenever the System Programmer is allowed to modify, recompile, and rebuild the TCB, strict configuration management procedures should be followed at the installation site and evidence be gathered to demonstrate to the Accreditor that the system rating is maintained properly. Again, it should be noted that any modification to the evaluated TCB code or configuration may invalidate the system's rating.

The distinction between various Operator's and Administrator's functions are established by:

(1) who performs the system configuration, distribution, installation and maintenance,

(2) who defines the user and the system security characteristics,

(3) who performs systems operations such as routine maintenance and response to user requests. This section recommends a more structured separation of roles that provides more effective management of the computer resources and accountability for those personnel.

5.1. FUNCTIONS OF THE SECURITY ADMINISTRATOR

The security-relevant functions of the Security Administrator can operate at more than one security level, and invoke processes or programs that operate with some system privileges. Thus, these functions must be trusted to a high degree. These functions include identification and authentication functions, mandatory access control functions, and discretionary access control functions.

1. The identifica*tion and authentica*tion functions of the Security Administrator may include:

The setting of the parameters of the login/out mechanism, such as:

* timeout period (maximum amount of time the system waits for the next command or for the completion of the current command);

* maximum login time (maximum amount of time the user may remain logged in to a system);

* limit of successive, unsuccessful tries to log in from a specific terminal before Administrators are notified;

* limit of successive, unsuccessful tries to log in to an account, regardless of the terminal location, before Administrators are notified;

* terminal lockout establishment and resetting;

* multiple (simultaneous) login attributes;

* whether a specific user's login needs to trigger an administrative warning (to the Administrator or to the Operator's console).

The setting of the authentication parameters; the Security Administrator functions may include those that carry out the following decisions:

* if the authentication mechanism is password-based, the Security Administrator determines the password characteristics (whether the user's password choice is user-generated or system-generated, the setting of the minimum and maximum password age, the password complexity parameters, etc.);

* if the mechanism is dialogue-based, the Administrator installs the dialogue programs on a per-user basis;

* the Administrator defines and manages the distribution of special passwords for the trusted processes that are started by passwords (i.e., the TCB repair and maintenance processes, such as security-label repair, etc.).

[Note: The above decisions are made when the system is installed for a particular organization, and the system Security Administrator carries out the installation decisions made by that organization.]

The definition of user account and registration profile; this definition may include:

* user identifier (this should be unique for the lifetime of the system); initial user password; change of user password;

* user's full name, address, and affiliation;

* user's group identifiers (these should also be unique for the lifetime of the system);

* user's default group.

The definition of group accounts and registration profile; this definition may include:

* user group id (this should be unique for the system's lifetime);

* group title, group administrator identifier, name and address;

* group disk quota;

* group statistics.

[Note: In some environments, the user and the group identifiers of registered users may not be disclosed to other users. Note also that, whenever the TCB does not automatically create unique identifiers for users and for groups, the system Security Administrator does not reuse user/group names until he is certain that name conflicts do not occur.]

2. The mandatory access control functions of the Security Administrator may include the following:

Definition and maintenance of the security label map; this includes functions such as the mapping between internal representations and human-readable representations of security lables.

Setting of the security-level limits and the default security levels for: the system, the users, the user groups, the system devices, and the file systems.

Labeling of imported unlabeled data, and unlabeled media such as disk packs.

Reclassification of objects; this includes:

* object upgrade or downgrade;

* label overrides on user output;

* restoration of damaged labels (whenever this function is not provided by the System Programmer role).

3. The discretionary access control functions of the Security

Administrator may include the following:

Initialization of the discretionary access privileges for group administrators to group directories and group devices; also, initialization of storage quotas for user groups.

Definition and maintenance of group membership (whenever special group administrators are not supported).

[Note: Since any change in group membership affects all discretionary access control decisions made by individual users, such changes should not take place without prior consultation with the users who may be affected by this decision.]

Setting of discretionary privileges on file systems.

Changes of object ownership in systems that support the notion of ownership; also, changes of discretionary privileges on objects whose privileges are accidentally deleted by the object's creators or owners.

Discretionary distribution, review, and revocation of privileges on behalf of object creators/owners in systems that do not allow individual users to distribute, review, and revoke privileges directly (i.e., where the control of object sharing is centralized [9]).

4. Additional functions of the Security Administrator are listed below. Specifically, the Security Administrator may:

Perform consistency checks to verify that:

* the database of user and system security profiles satisfies the system security requirements and is in a consistent state;

* the TCB is installed properly (e.g., displays and checks installation tables);

* the TCB does not contain extraneous programs (e.g., programs that are privileged but are not part of the TCB configuration).

Determine that the current system configuration is within the constraints established by configuration management and the System Programmer. This includes the verification of:

* device and terminal registration;

* maximum storage size;

* file (device) system name table and file (device) mount tables;

* device and terminal connection database.

Cut off user/group accounts (whenever the Account Administrator is not defined as a separate role).

Delete user/group accounts.

Display and update constants of various system tables.

Initiate and analyze the system integrity tests.

Supervise the maintenance procedures (hardware, etc.).

Respond to real-time alarm messages (B3 and higher).

Destruction of errant processes.

Definition of the site identifier, logo, and the site authentication protocols within a network.

Set up and access the following four types of databases:

* The database of the user and system security profiles;

* The security label map;

* The file system hierarchy;

* The system configuration database [this includes the current hardware configuration and the security-level limits of the various devices, terminal connections, the file-system name and mount database, etc.].

All the modifications to these databases are performed by the Security Administrator using the commands of a trusted database editor and the system's trusted path. Although the trusted path mechanism is not required for these modifications in class B2 systems, the trusted editor commands are part of the administrative interface commands that must be supported by all trusted systems. All actions of the Security Administrators are audited.

5.2. FUNCTIONS OF THE SECURE OPERATOR

The security-relevant functions of the Operator role can operate across more than one security level and sometimes invoke processes that require system privileges. Thus, these functions require a high degree of trust. An Operator who executes security-relevant operations is called the Secure Operator. These functions of the Secure Operator may include the following:

1. Booting and shutting down the system; setting the system's clocks; also, setting the security level of individual system devices within the range of levels allowed by the Security Administrator's database.

[Note: Shutting down the system requires that the Operator ensure that appropriate physical and administrative security features be in place to protect the information while the system is not running. For example, shutting down for maintenance might require that the date be removed and the system cleared.]

2. Locating damaged user files and volumes. The "salvager"

process identifies damaged labels (e.g., labels inconsistent with those of containing directories and files) and deletes all access to the corresponding objects until repair is finished by the System Programmer and Security Administrator.

3. Performing routine maintenance of TCB databases.

The Operator performs the following routine maintenance operations:

* audit file backup (whenever this is not included in the Auditor's role);

* security-level changes for some devices (these are within the limits set by the system Security Administrator);

* user database backup;

* security-map backup;

* TCB tables backup.

It must be noted that the Operator should not have the privilege to modify file contents for file backup.

4. Performing on-line terminal and device tests (including authentication tests).

5. Responding to user's requests.

The Operator should be able to respond to the following user requests:

* mount/unmount physically (externally) labeled removable media (e.g., tape reels and disk packs);

* import/export other physically (externally) labeled data into/from the system.

It must be noted that all Operator's actions must be auditable. Mounting unlabeled storage devices is not recommended. The TCB needs the Label information in order to correct access control decisions. If the Operator is not provided the label, the system will not be able to enforce the policy correctly.

5.3. FUNCTIONS OF THE ACCOUNT ADMINISTRATOR

The security-relevant functions of the Administrator role may not need the special privileges to operate properly, but in most installations they will be trusted processes However, all output generated by the Account Administrator will be marked with the highest security level. Otherwise, leakage of classified information may take place (e.g., encoded in the user bills). The nonsecurity-relevant role of the Security Administrator is called the Account Administrator.

The (nonsecurity-relevant) functions of the Account Administrator are listed below. Specifically, the Account Administrator:

1. Installs and maintains accounting files.

2. Turns system accounting on and off.

3. Runs accounting tools and produces accounting reports/bills.

4. Enables and disables accounts at users' requests (whenever this function is not provided by the Security Administrator); however, the Account Administrator does not have the privilege to define or change the users' security profiles.

5. Establishes the billing rates, prices and policies.

6. On a regular basis, collects system statistics such as:

* system availability;

* system configuration;

* disk/CPU/memory statistics.

7. Publishes revenue/cost reports.

5.4. FUNCTIONS OF THE AUDITOR

The Auditor role invokes processes that operate with system privileges. Thus, all functions of the Auditor require a high degree of trust. These functions include those that enable the audit selectivity mechanism (e.g., audit-event setup and change), the management of audit trails, the setting of the covert-channel delays and randomization variables, audit data compression and postprocessing analysis [7]. Data generated by the Auditor must be classified at the System High level since they may contain information generated at all security levels defined in the system. System High is defined as the security label that dominates all other security labels in the system. In a sense, it is the highest possible label. It would be beneficial, and possibly necessary, to create the System High level such that it is hierarchically higher than all the data levels used in the system. This approach has the benefit that the mandatory access controls provide additional protections for the audit data since only the Auditor would have authorization for this level.

1. The Auditor functions that define the events recorded in the audit log (or trail) may include:

Functions that turn on and off events that should be recorded in the audit trail to ensure the consistency of subsequent events selected by the Auditor. These events ensure that the postprocessing tools function properly. For example, in systems where object-unique names are represented by file system pathnames, any change to the working directory relative to which pathnames are interpreted, should be audited. (An object-unique name is the unique name that identifies and distinguishes a particular object from all other objects in a system. In a hierarchical file system, the object-unique name includes the associated directory names so users can use the same name for objects in different directories). Otherwise, audit analysis tools that read audit events recorded after a directory change cannot identify objects unambiguously. For similar reasons, all events that record process creation or destruction and identification or authentication actions should be selected whenever the audit is on.

Functions that display all security-relevant events which can be audited.

(The determination of the security-relevant events in a system is done at design time, and is based on the interpretation of the chosen security policy and accountability models in the system. Any event, such as those provided by a user invocation of a TCB or trusted process call, is security-relevant if it causes a state transition or if it denies a state transition in the model's interpretation. For example, the introduction of an object in an address space of a process is security-relevant in a system designed to support the Bell-LaPadula model because it causes a state transition in the interpretation of the current-access-set component of that model's interpretation [2]. Similarly, distribution and revocation of access privileges cause a state transition because they modify the access-matrix component of the model; whereas a change in security level of an object/subject causes a state transition because it modifies the security-function component of the model. Other state transitions, which should also be audited, may modify multiple components of a system state; e.g., the creation/destruction of objects that modify both the object hierarchy and the access matrix. Additional security-relevant events may be derived from the interpretation of the trusted facility management model whenever such a model is not included in the security policy model. Also, additional security-relevant events may be derived from the covert-channel handling requirements of the TCSEC).

Functions that turn on or off audit events selectively on a per-user, per-process, per-security-level or per-object basis are also included here. These events may be signaled by the processors, TCB, or trusted processes. Selection of auditor-determined subsets of these events should also be possible.

Functions that turn on or off events representing accumulations of other auditable events (e.g., multiple successive unsuccessful logins) and alarms are also included here.

2. Auditor functions that help manage the audit files mayinclude:

Creation and destruction of audit logs and postprocessing audit files.

Change of audit-log size and of warning points. The warning points may be expressed as a specific number, or percentage, of bytes available in the audit log. When these warning points are crossed by the event recording mechanism, an auditor warning may be given by the system.

If the audit log becomes full and the audit mechanism is on, then the system may stop and delay further activity until the Auditor takes corrective action [7].

Functions used to empty full audit files.

Functions that format and compress events in the audit log and postprocessing audit files. The formatting functions may convert binary audit data into text format, and combine partial event records into the required record format. The storing of formatted postprocessing files may require the use of compression techniques to improve storage utilization.

Functions that display the audit log and postprocessing audit files in various formats.

Consistency checking functions which operate on the entire auditor database for use after system crashes.

3. Functions that set the delays or the randomization values forcovert channel handling should also be included in the Auditor's role. The reason for this is that the covert channel handling guideline of the TCSEC correlates the covert-channel audit requirement with specific covert-channel bandwidth values and, therefore, with delay values and randomization ranges. For example, depending upon the values set for the audit delays, specific channels may, or may not, need to be audited. Thus, the specification of the delay values and randomization ranges becomes the duty of the Auditor. These functions may include:

The setting of the default and current values of the delays for single covert channels or for groups of covert channels.

The setting of the default and current values of the randomization ranges for covert channels arising from the dynamic allocation or deallocation of indices in TCB tables.

4. Functions that perform the postprocessing of the audit data are necessary for any audit log analysis and, therefore, should be included in any trusted system. Although some of these functions are independent of the required audit analysis, such as the functions that retrieve various fields of the audit logs, most of these functions are specific to the postprocessing analysis required by specific applications.

In summary, the functions of the Auditor role may set up, access and modify the following types of databases:

* audit log files containing full or partial records of audit events in binary or text formats;

* audit event file containing the definition of all auditable events in the system;

* selected-event file containing the definitions of all events selected on a per-user, per-process, per-security-level, per-object basis;

* formatted or compressed audit files containing the input to the postprocessing phase;

* audit report files.

Access to the audit databases may be performed only by individuals who can assume the Auditor role, using the commands defined for that role. Use of Auditor commands must be audited. For class B3 and above systems, the use of Auditor commands must be through the trusted path mechanism.

5.5. FUNCTIONS OF THE OPERATOR

The security-relevant functions of the Operator role do not need all the system privileges to operate properly. However, the Operator should be able to change the authorization of his processes between System Low and System High because he may need to operate at different security levels. System Low is the security label that is dominated by all other security labels in the system. In a sense, it is the lowest possible label.

The (security-relevant) functions of the Operator are defined below. Specifically, the Operator:

1. Performs user volume backup. This includes:

* complete volume dumps;

* complete volume retrievals.

2. Performs system performance metering.

3. Responds to various other user requests (request for the installation of user-level software packages, etc.).

4. Adjusts resource quotas for user-visible resources.

5.6. FUNCTIONS OF THE SYSTEM PROGRAMMER

The functions of the System Programmer role are the most security-sensitive functions of the system. They may affect the TCB configuration, distribution and maintenance. These functions are not necessarily audited and, thus, any error, omission, or malicious act, which affects the security of the entire system, may remain undetected. (However, some form of auditing, possibly off-line, is still necessary in some environments. Multiple Systems Programmers checking each others' actions may also be required in some environments for the execution of the System Programmer functions. Furthermore, a two-person rule may be instituted or built into the login procedure requiring that a System Programmer may not log in successfully unless another System Programmer is also logging in). Thus, the System Programmer functions should have the highest degree of trust in the system. The System Programmer functions may include the following:

1. Trusted system distribution; for example, this includes the generation and handling of the site's system master copy.

2. Setting of system configuration parameters (as specified by the site's configuration management policy); for example, this includes:

* generic system configuration;

* initialization of the TCB data structures (before any security profiles or audit characteristics are defined);

* loading of the TCB.

3. Nonroutine TCB maintenance; for example, this includes:

* analysis of dumps;

* installation of "patches" to the TCB code and data (for this the Operator should be able to recompile TCB code from modified source code and should use a trusted loader to reload the system);

* trusted recovery actions after system crashes; for example the Operator performs consistency checks on the file system structure, on individual TCB files, directories and tables, repairs damaged labels;

* repairs damaged security labels whenever this function is not provided by the Security Administrator role (damaged labels identified by Secure Operators or Users).

The databases of the System Programmer include:

* all TCB files (e.g., TCB code, security-map, auditor files);

* all TCB tables (e.g., interrupt vectors, trap tables, gates).

5.7. OTHER ROLES

Other administrative roles can be defined in a secure system. For example, in certain environments the role of the Analyst can be defined. An Analyst may be an otherwise unprivileged user who is trusted to label imported data from various system inputs, to create new files and label them as he sees fit. The Analyst cannot label any data file with a security level higher than his maximum clearance. All the Analyst's actions are audited as are those of a normal user.

When a system is tied into a network, additional roles may be necessary to ensure consistency and accuracy of the network policy enforcement. Such roles could involve additional security-relevant databases.

5.8. RELATIONSHIP AMONG ADMINISTRATIVE ROLES

The fine-grained separation of administrative roles defined above permits the establishment of a hierarchical relationship among administrative roles based on a notion of "role dominance" (not to be confused with the notion of dominance among security or integrity levels). This notion signifies the ability of an administrative user in a certain role to change the attributes of objects and security profiles of users in other roles and, if necessary, to log in and take actions in that role.

Object attributes include:

* access privileges;

* size;

* security and integrity levels; and

* ownership.

Profile attributes include:

* user and group identifiers;

* passwords;

* group membership; and

* time restrictions on user activity.

The above notion of role dominance can be useful because it provides both a measure of necessary trust (based on skills, on checking administrative users' background and interests, etc.) that should be invested in a role and a measure of vulnerability associated with that role. The most privileged role is that of the System Programmer. It dominates all other roles in the system and, consequently, it exhibits the highest degree of vulnerability. The Auditor role should be strictly separated from all other remaining roles defined in the system because it maintains sensitive information describing the behavior of all users, including the administrative ones. The Security Administrator dominates the Secure Operator, Account Administrator, Analyst, and user roles; however, the dominated roles are separated from each other. It must be noted that users in the same role do not dominate each other. Although they share most functions, privileges, and databases of the common role, their security profiles are disjoint to allow individual accountability. This helps distinguish the activities of individual users in the same role. Figure 4 illustrates the relationship among the administrative roles defined above. The system could provide a "flat" set of roles, functions and privileges, and the role relationships that could be managed administratively. Implementations of hierarchical relationships among administrative roles can benefit from the use of mandatory security and, especially, integrity models. Mandatory integrity models, such as the Biba model [4] and the Clark-Wilson model [8], could be used to guide the design of the above-mentioned roles and hierarchical relationships, as discussed below.

6. IMPACT OF OTHER TCSEC REQUIREMENTS ON TRUSTED FACILITY MANAGEMENT

The major areas of the TCSEC requirements (security policy, accountability, assurance and documentation) impact on trusted facility management. The design and implementation of the functions of various administrative roles may use some of the security mechanisms and policies of the underlying system to implement some of their special protection requirements or may choose to implement new protection mechanisms and policies. For example, the implementors of Security Administrator functions may use the discretionary access control mechanisms or may choose to implement to protect the Security Administrator databases from other administrative users and from normal users. This section examines the relationship of other TCSEC requirements to trusted facility management.

6.1. SECURITY POLICY

To support the system's security policies, the functions of trusted facility management must control access to, and sharing of, administrative data. Trusted processes implementing the security functions of the Administrator's and Operator's role share files of the administrative database in a variety of ways. Some files are private to each role and are never shared with other roles, with other users of the same role, or with nonadministrative trusted processes. For example, the security label map file is private to the Security Administrator role, the audit log and the postprocessing audit files are private to the Auditor role, and the accounting files are private to the Account Administrator role. All such files are shared among all users of the same role. Other files, such as those containing the user and group registration, may be shared between processes of different roles. These files may be read and written by Security Administrator processes, and are read by Auditor, Secure Operator and Account Administrator processes. Account Administrators and Operators may perform special tasks, such as the collection of user and system statistics and performance metering, for which they would create and maintain private files (those not shared with others in the same role). Furthermore, other files are shared between processes of an administrative role and nonadministrative trusted processes. For example, the user password file is read and written by the Security Administrator role, read by the "login" trusted process, and read and written by the "change-password" trusted process, which can be invoked by any user.

To control access to administrative data and to implement the above-mentioned sharing relationships among processes of the administrative roles, the design and implementation of trusted facility management may, or may not, rely on discretionary and mandatory access controls of the underlying system. If they do, some processes implementing role functions, which need to read and write files at all security levels (e.g., Accounting, Auditor, and Secure Operator processes), would need to bypass the mandatory access controls at least occasionally. Some other processes will operate at the highest level in the system (e.g., accounting and audit processes) and maintain data files at this level (e.g., audit log and postprocessing files, accounting files).

Whenever the sharing relationships among programs and processes of the administrative roles cannot be supported by existing mechanisms, new mechanisms have to be introduced. For example, the association of specific programs implementing administrative functions with roles may require the implementation of restricted command processors, of restricted groups that cannot be modified by the Security Administrator, or of other more complex integrity mechanisms (discussed below). In all such cases, the design and implementation of trusted facility management functions should follow existing guidelines (see example,[9]).

6.2. ACCOUNTABILITY

The accountability requirements of the TCSEC impose several constraints on the implementation of trusted facility management, in addition to the separation of roles. First, the identification and authentication of all administrative users must be unambiguous, and must be done on an individual basis, not on a per-role basis. For example, if all users of a role share the same password, accountability will be lost since any user can take the identity of other users of the same role and commit acts of intrusion attributable to those users.

Second, the trusted-path mechanism for classes B3 and above must ensure that the administrative users are connected to the commands or processes that belong to their role, and that no other users or processes can interpose themselves into the path connecting any combination of the administrative users, their commands, and their processes. This can be accomplished by providing administrative consoles recognized and separated by the TCB hardware or software from the rest of the terminals, or by the design of a full (i.e., B3-A1) trusted-path mechanism.

Third, use of all administrative functions, other than those used by System Programmers in maintenance mode, must be audited. This implies that trusted programs and processes implementing these commands should be able to request the writing of audit records during the execution of those commands. In all areas of accountability, the design and implementation of trusted facility management functions should follow existing guidelines (see example, [7]).

6.3. ASSURANCE

The assurance requirements of the TCSEC have a significant impact on trusted facility management both in the operational and in the life-cycle areas. These requirements affect both the design and the implementation of the trusted facility management functions.

6.3.1. Operational Assurance

The only relevant areas of operational assurance are the system architecture and the trusted recovery areas. The covert channel analysis area is not relevant here because (1) all users in security-relevant administrative roles have been screened for this position of trust and are therefore expected not to disclose information in an unauthorized way, and (2) all code implementing administrative functions is reviewed to ensure, to the largest possible extent, that no Trojan horses are present. The system integrity requirements of the TCSEC are also irrelevant here as they deal only with the test of proper hardware and firmware operation.

The system architecture requirements impose major constraints on the design of trusted facility management. Because all the security-relevant and accountability functions of the administrative roles are part of a system's TCB, all requirements of TCB interface definition apply to the administrative interfaces. Similarly, all requirements of internal TCB structuring, such as those of modularity, abstraction, information hiding, and layering apply to the design and implementation code of the programs and processes of trusted facility management. Careful analysis and documentation of this design and implementation area, as well as careful scrutiny by evaluators, is expected in this area.

The application of the least privilege principle to the design of trusted processes is also required of the administrative processes of the TCB. Several specific design requirements should be observed here. First, the protection of the administrative databases should be performed at the granularity of individual files (or segments) and individual privileges. (The term file is used here in a generic sense to represent a logically small structure such that the structure does not include information unrelated to the specific function). Second, programs and processes of the administrative roles should have access only to the TCB and user files, and to the privileged TCB calls, that are necessary for implementing those roles, but to no other files or calls. Several design alternatives are available in this area. For example, certain files should be associated only with certain processes. Privileged TCB calls, which can be represented by ring-gate descriptors [15,19], domain-entry capabilities [13], or per-process privilege vectors corresponding to specific calls [16,14], should be associated with processes only on an "as needed" basis. These associations can be controlled by careful application of nondiscretionary labels and authorizations at system configuration or installation time.

The only specific requirement of trusted recovery imposed on the design and implementation of trusted facility management is that the consistency of the administrative databases be maintained after system crashes. This requirement can be satisfied by ensuring that :

* these databases are stored on nonvolatile storage that survives system crashes;

* that updates to such storage are atomic ;

* that at least one of the administrative roles is equipped with commands for checking the consistency of the administrative file contents. Note that this could be a fully automated mechanism not requiring administrator interaction.

6.3.2. Life-Cycle Assurance

Most life-cycle assurance requirements apply to the processes and interfaces of trusted facility management as stated. For example, security testing, configuration management, and trusted distribution requirements of the TCB apply to trusted facility management to the degree of rigor commensurate with the chosen evaluation class. This is the case because the TCB code and interfaces include the security-relevant code and interfaces of trusted facility management.

In contrast, only some of the requirements of the design specification and verification area apply to the trusted facility management directly. For example, the need for accurate DTLSs for the TCB interfaces applies as stated. However, the requirements for a formal model, for an interpretation of this model in the DTLSs of the trusted facility management part of the TCB, and for a convincing argument that the DTLSs are consistent with the model are not directly applicable here. The reason for this is that no generally acceptable formal model of the trusted facility management area exists to date. Should a generally acceptable formal model become available, then all requirements of the design specification and verification area would apply to trusted facility management directly.

6.4. DOCUMENTATION

The documentation requirements of the TCSEC relevant to the trusted facility management area are the trusted facility manual requirements in section 3, the test documentation requirements, and some of the design documentation [8]. In the design documentation area, only the requirements referring to the DTLSs, TCB internal structuring, and enforcement of the least privilege principle are relevant.

GLOSSARY

Access

A specific type of interaction between a subject and an object that results in the flow of information from one to the other.

Account Administrator

An administrative role or user assigned to maintain accounting files, tools, user accounts, and system statistics.

Administrative User

A user assigned to supervise all or a portion of an AIS.

Administrator

See Administrative User.

Approval/Accreditation

The official authorization that is granted to an AIS to process sensitive information in its operational environment, based upon comprehensive security evaluation of the system's hardware, firmware, and software security design, configuration, and implementation and of the other system procedural, administrative, physical, TEMPEST, personnel, and communications security controls.

Audit

To conduct the independent review and examination of system records and activities.

Audit Event Selection

Selection, by authorized personnel, of the auditable events that are to be recorded on the audit trail.

Audit Mechanism

The processes used to collect, review, and/or examine system activities.

Audit Postprocessing

Processing, under the control of authorized personnel, of specified events that had been recorded on the audit trail.

Audit Trail

A chronological record of system activities that is sufficient to enable the reconstruction, reviewing, and examination of the sequence of environments and activities surrounding or leading to an operation, a procedure, or an event in a transaction from its inception to final results.

Auditable Event

Any event that can be selected for inclusion in the audit trail. These events should include, in addition to security-relevant events, actions taken to recover the system after failure and any events that might prove to be security-relevant at a later time.

Auditor

An authorized individual, or role, with administrative duties, which include selecting the events to be audited on the system, setting up the audit parameters which enable the recording of those events, and analyzing the trail of audit events.

Authenticate

(1) To verify the identity of a user, device, or other entity in a computer system, often as a prerequisite to allowing access to resources in a system.

(2) To verify the integrity of data that have been stored, transmitted, or otherwise exposed to possible unauthorized modification.

Authenticated User

A user who has accessed an AIS with a valid identifier and authentication combination.

Auto*mated Informa*tion System (AIS)

An assembly of computer hardware, software and/or firmware configured to collect, create, communicate, compute, disseminate, process, store, and/or control data or information.

Bandwidth

A characteristic of a communication channel that is the amount of information that can be passed through it in a given amount of time, usually expressed in bits per second.

Category

A restrictive label that has been applied to classified or unclassified data as a means of increasing the protection of the data and further restricting access to the data.

Channel

An information transfer path within a system. May also refer to the mechanism by which the path is effected.

Covert Channel

A communication channel that allows a process to transfer information in a manner that violates the system's security policy. See also: Covert Storage Channel, Covert Timing Channel.

Covert Storage Channel

A covert channel that involves the direct or indirect writing of a storage location by one process and the direct or indirect reading of the storage location by another process. Covert storage channels typically involve a finite resource (e.g., sectors on a disk) that is shared by two subjects at different security levels.

Covert Timing Channel

A covert channel in which one process signals information to another by modulating its own use of system resources (e.g., CPU time) in such a way that this manipulation affects the real response time observed by the second process.

Data

Information with a specific physical representation.

Data Integrity

The property that data meet an a priori expectation of quality.

Descriptive Top-Level Specification (DTLS)

A top-level specification that is written in a natural language (e.g., English), an informal program design notation, or a combination of the two.

Discretionary Access Control

A means of restricting access to objects based on the identity and need-to-know of the user, process and/or groups to which they belong. The controls are discretionary in the sense that a subject with a certain access permission is capable of passing that permission (perhaps indirectly) on to any other subject.

Formal Security Policy Model

A mathematically precise statement of a security policy. To be adequately precise, such a model must represent the initial state of a system, the way in which the system progresses from one state to another, and a definition of a "secure" state of the system. To be acceptable as a basis for a TCB, the model must be supported by a formal proof that if the initial state of the system satisfies the definition of a "secure" state and if all assumptions required by the model hold, then all future states of the system will be secure. Some formal modeling techniques include: state transition models, temporal logic models, denotational semantics models, and algebraic specification models.

Formal Top-Level Specification (FTLS)

A Top-Level Specification that is written in a formal mathematical language to allow theorems showing the correspondence of the system specification to its formal requirements to be hypothesized and formally proven.

Functional Testing

The segment of security testing in which the advertised features of a system are tested, under operational conditions, for correct operation.

Least Privilege

The principle that requires that each subject be granted the most restrictive set of privileges needed for the performance of authorized tasks. The application of this principle limits the damage that can result from accident, error, or unauthorized use.

Mandatory Access Control

A means of restricting access to objects based on the sensitivity (as represented by a label) of the information contained in the objects and the formal authorization (i.e., clearance) of subjects to access information of such sensitivity.

Multilevel Device

A device that is used in a manner that permits it to process data simultaneously of two or more security levels without risk of compromise. To accomplish this, sensitivity labels are normally stored on the same physical medium and in the same form (i.e., machine-readable or human-readable) as the data being processed.

Multilevel Secure

A class of system containing information with different sensitivities that simultaneously permits access by users with different security clearances and needs-to-know, but prevents users from obtaining access to information for which they lack authorization.

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.

Object-Unique Names

The unique name that identifies and distinguishes a particular object from all other objects in a system. In a hierarchical file system, the object-unique name includes the associated directory names so users can use the same name for objects in different directories.

Operator

An administrative role or user assigned to perform routine maintenance operations of the AIS and to respond to routine user requests.

Output

Information that has been exported by a TCB.

Password

A protected/private character string that is used to authenticate an identity.

Process

A program in execution. It is completely characterized by a single current execution point (represented by the machine state) and address space.

Read

A fundamental operation that results only in the flow of information from an object to a subject.

Read Access (Read Privilege)

Permission to read information.

Secure Operator

An administrative role (or user) assigned to perform those aspects of the Operator role that can affect the security relevant data used by the TCB to enforce its policy (e.g., notifying the TCB of the security label of a newly mounted tape).

Security Administrator

An administrative role (or user) responsible for the security of an Automated Information System and having the authority to enforce the security safeguards on all others who have access to the Automated Information System (with the possible exception of the Auditor). Also called System Administrator.

Security Label Map

A map defining the correspondence between the binary and ASCII formats of security levels (e.g., between binary format of security levels and sensitivity labels).

Security Level

The combination of a hierarchical classification and a set of nonhierarchical categories that represents the sensitivity of information.

Security Policy

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

Security Policy Model

A formal presentation of the security policy enforced by the system. It must identify the set of rules and practices that regulate how a system manages, protects, and distributes sensitive information.

Security-Relevant Event

Any event that attempts to change the security state of the system, (e.g., change discretionary access controls, change the security level of the subject, change user password, etc.). Also, any event that attempts to violate the security policy of the system (e.g., too many attempts to login, attempts to violate the mandatory access control limits of a device, attempts to downgrade a file, etc.).

Security Testing

A process used to determine that the security features of a system are implemented as designed. This includes hands-on functional testing, penetration testing, and verification.

Sensitive Information

Information that, as determined by a competent authority, must be protected because its unauthorized disclosure, alteration, loss, or destruction will at least cause perceivable damage to someone or something.

Sensitivity Label

A piece of information that represents the security level of an object and that describes the sensitivity (e.g., classification) of the data in the object. Sensitivity labels are used by the TCB as the basis for mandatory access control decisions.

Separation of Privilege

The separation of functions, namely between the commands, programs, and interfaces implementing those functions, such that malicious or erroneous code in one function is prevented from affecting the code or data of another function.

Spoofing

An attempt to gain access to a system by posing as an authorized user. Also called masquerading or mimicking.

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.

Subject Security Level

A subject's security level is equal to the security level of the objects to which it has both Read and Write access. A subject's security level must always be dominated by the clearance of its associated user.

System Administrator

See Security Administrator.

System High

The security label that dominates all other security labels in the system. In a sense, it is the highest possible label.

System Low

The lowest security level supported by a system at a particular time or in a particular environment.

System Programmer

An administrative role (or user) responsible for trusted system distribution, configuration, installation, and nonroutine maintenance.

Top-Level Specification (TLS)

A nonprocedural description of system behavior at the most abstract level; typically, a functional specification that omits all implementation details.

Trap Door

A hidden software or hardware mechanism that can be triggered to permit system protection mechanisms to be circumvented. It is activated in some innocent-appearing manner; e.g., a special "random" key sequence at a terminal. Software developers often introduce trap doors in their code to enable them to reenter the system and perform certain functions. Synonymous with back door.

Trojan Horse

A computer program with an apparently or actually useful function that contains additional (hidden) functions that surreptitiously exploit the legitimate authorizations of the invoking process to the detriment of security or integrity.

Trusted Computer System

A system that employs sufficient hardware and software assurance measures to allow its use for processing simultaneously a range of sensitive or classified information.

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 enforce correctly a unified 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 level) related to the security policy.

Trusted Path

A mechanism by which a person at a terminal can communicate directly with the Trusted Computing Base. This mechanism can only be activated by the person or the Trusted Computing Base and cannot be imitated by untrusted software.

User

Person or process accessing an AIS either by direct connections (i.e., via terminals), or indirect connections (i.e., prepare input data or receive output that is not reviewed for content or classification by a responsible individual).

Verification

The process of comparing two levels of system specification for proper correspondence (e.g., security policy model with top-level specification, top-level specification with source code, or source code with object code). This process may or may not be automated.

Virus

A self-propagating Trojan horse, composed of a mission component, a trigger component, and a self-propagating component.

Vulnerability

A weakness in system security procedures, system design, implementation, internal controls, etc., that could be exploited to violate system security policy.

Write

A fundamental operation that results only in the flow of information from a subject to an object.

Write Access (Write Privilege)

Permission to write an object.

REFERENCES

1. Baldwin, R. W., "Rule-Based Analysis of Computer Security," Technical Report MIT/LCS/TR-401, March 1988.

2. Bell, D.E., and L. J. LaPadula, "Secure Computer System: Unified Exposition and Multics Interpretation," MITRE Corp., Rep. No. MTR-2997, 1976 (available as NTIS AD-A023588).

3. Biba, K.J., "Integrity Considerations for Secure Computer Systems," Mitre Corp., MTR-3153, Bedford, Mass., June 1975.

4. Bishop, M., "How to Write a Setuid Program"; login, vol. 12, no. 1, January/February 1987.

5. Bishop M., "Managing Superuser Privileges under Unix," Research Institute for Advanced Computer Science, Technical Report, NASA Ames Research Center, Moffet Field, California, (June 1986).

6. Clark, D.D., and D. R. Wilson, "A Comparison of Commercial and Military Computer Security Policies," Proc. of the IEEE Symp. on Security and Privacy, Oakland, Calif., April 1987.

7. Department of Defense -- A Guide To Understanding Audit In Trusted Systems, NCSC-TG-001, version-1, July 1987.

8. Department of Defense -- A Guide To Understanding Design Documentation In Trusted Systems, NCSC-TG-007, version-1, October 1988.

9. Department of Defense, A Guide to Understanding Discretionary Access Control in Trusted Systems, NCSC-TG-003, version-1, September 1987.

10. Department of Defense, Password Manage*ment Guideline, CSC-STD-002-85, April 1985.

11. Department of Defense, Trusted Computer System Evaluation Criteria, DoD 5200.28-STD, December 1985.

12. Gligor V.D., C.S. Chandersekaran, R.S. Chapman, L.J. Dotterer, M.S. Hecht, W.D. Jiang, A. Johri, G.L. Luckenbaugh, and N. Vasudevan, "Design and Implementation of Secure Xenix," IEEE Trans. on Software Engineering, vol. SE-13, No. 2, February 1986.

13. Gligor V. D., J.C. Huskamp, S.R. Welke, C. J. Linn, and W.T. Mayfield, "Traditional Capability-Based Systems: An Analysis of their Ability to Meet the Trusted Computer Security Evaluation Criteria," Institute for Defense Analyses, IDA Paper P-1935, February 1987

14. Hecht M.S., M.E. Carson, C.S. Chandersekaran, R.S. Chapman, L.J. Dotterer, V.D. Gligor, W.D. Jiang, A. Johri, G.L. Luckenbaugh, and N. Vasudevan, "Unix Without the Superuser," Proc. of the Usenix Conference, Phoenix, Arizona, June 1987.

15. Intel Corp., iAPX 286 Programmers Reference Manual, Chapter 7, section 5, Intel Corp., 1983.

16. Knowles, F., and S. Bunch, "A Least Privilege Mechanism for Unix," Proc. of the 10th Na*tional Computer Security Conference, Baltimore, MD., September 1987.

17. Lee, T.M.P., "Using Mandatory Integrity to Enforce 'Commercial' Security," Proc. of the IEEE Symp. on Security and Privacy, Oakland, Calif., 1988.

18. Saltzer, J. H., and M. D. Schroeder, "The Protection and Control of Information Sharing in Computer Systems," Proc. of the IEEE, vol. 63, no. 9, September 1975.

19. Schroeder, M.D. and J.H. Saltzer, "A Hardware Architecture for Implementing Protection Rings," Communica*tions of the ACM, vol. 15, no. 3, March 1972.

20. Thompson, K., "Reflections on Trusting Trust," Turing Award Lecture, Communications of the ACM, vol. 27, no. 8, August 1984.