NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

MANDATORY SECURITY POLICY ENFORCEMENT IN COMMERCIAL OFF THE SHELF DATABASE MANAGEMENT SYSTEM SOFTWARE: A COMPARATIVE ANALYSIS OF INFORMIX ON-LINE/SECURE AND TRUSTED ORACLE

by

Keith E. Muschalek

September 1994

Co-Advisors: Cynthia Irvine
C. Thomas Wu

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	September 1994	Master's Thesis

4. TITLE AND SUBTITLE
MANDATORY SECURITY POLICY ENFORCMENT IN COMMERCIAL OFF THE SHELF DATABASE MANAGEMENT SYSTEM SOFTWARE: A COMPARATIVE ANALYSIS OF INFORMIX-ONLINE/SECURE AND TRUSTED ORACLE (U)

5. FUNDING NUMBERS

6. AUTHOR(S)
Muschalek, Keith Edward

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/ MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/ MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The objective of this thesis is to analyze the mandatory, access control (MAC) features of two commercial multilevel trusted database management systems (DBMS): Trusted ORACLE 7 and Informix-OnLine/Secure 5.0. We are attempting to determine how the problem of multilevel sharing of information is addressed in each multilevel secure DBMS.

Commercially available documentation is used to examine the mandatory access controls enforced on labeled subjects and labeled objects and to compare them to the Class B1 requirements for MAC and labeling set forth in the Trusted Computer System Evaluation Criteria (TCSEC). A decomposition of the TCSEC requirements for MAC and labeling is mapped to the DBMS documentation to determine if the Class B1 requirements are met by each DBMS. With the TCSEC mapping as a reference, the interface features in support of MAC are analyzed and compared between the products.

This analysis shows that each DBMS uses different schema objects and privilege sets to enforce its mandatory security policy. The MAC mechanism of each product is based on the Bell-LaPadula security model, extended to prohibit the writeup of data from lower level subjects to higher level objects. Each DBMS allows traditional trusted subjects to writedown data. When special privileges are granted to users, readups and writeups are permitted in both DBMSs.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Database Security, Multilevel Secure Database Management Systems, B1 DBMS, TCSEC analysis, Database analysis, Database evaluations.	163
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

Approved for public release; distribution is unlimited

MANDATORY SECURITY POLICY ENFORCEMENT IN COMMERCIAL OFF THE SHELF DATABASE MANAGEMENT SYSTEM SOFTWARE: A COMPARATIVE ANALYSIS OF INFORMIX ON-LINE/SECURE AND TRUSTED ORACLE

by

Keith Edward Muschalek
Captain, United States Army
B.B.A., Texas A&I University, 1987

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

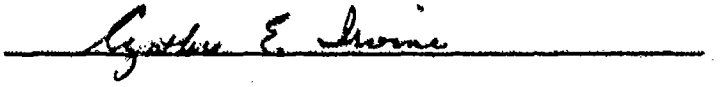
from the

NAVAL POSTGRADUATE SCHOOL

September 1994

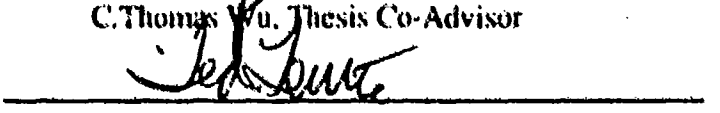
Author: 

Keith Edward Muschalek

Approved By: 

Cynthia Irvine, Thesis Co-Advisor

C. Thomas Wu, Thesis Co-Advisor



Ted Lewis, Chairman,
Department of Computer Science

ABSTRACT

The objective of this thesis is to analyze the mandatory access control (MAC) features of two commercial multilevel trusted database management systems (DBMS): Trusted ORACLE 7 and Informix-OnLine/Secure 5.0. We are attempting to determine how the problem of multilevel sharing of information is addressed in each multilevel secure DBMS.

Commercially available documentation is used to examine the mandatory access controls enforced on labeled subjects and labeled objects and to compare them to the Class B1 requirements for MAC and labeling set forth in the Trusted Computer System Evaluation Criteria (TCSEC). A decomposition of the TCSEC requirements for MAC and labeling is mapped to the DBMS documentation to determine if the Class B1 requirements are met by each DBMS. With the TCSEC mapping as a reference, the interface features in support of MAC are analyzed and compared between the products.

This analysis shows that each DBMS uses different schema objects and privilege sets to enforce its mandatory security policy. The MAC mechanism of each product is based on the Bell-LaPadula security model, extended to prohibit the writeup of data from lower level subjects to higher level objects. Each DBMS allows traditional trusted subjects to writedown data. When special privileges are granted to users, readups and writeups are permitted in both DBMSs.

TABLE OF CONTENTS

I. INTRODUCTION

The security of information within computer systems is a major issue for system automation professionals of the 1990's. The disclosure of sensitive information, the modification of valuable data files, and the disruption of service, by both authorized and unauthorized personnel, have plagued system administrators for many years. Today, with the advent of interactive network computing and the "information superhighway", information has to be protected more than ever.

Security issues have been a concern in the national security and defense establishments since the dawn of the computer age. National defense mandated major requirements for security in the development and acquisition of automated computer systems. Work by government personnel and defense contractors brought about the development of systematic criteria for measuring the effectiveness and trustworthiness of security mechanisms within computer systems. These "criteria" became the Trusted Computer Security Evaluation Criteria (TCSEC), commonly called the "Orange Book." The TCSEC is the metric by which the United States Government measures the security effectiveness of an automated computer system.[1]

This thesis will be an attempt to conduct a comparative analysis of selected security features of two leading U.S. database management system (DBMS) products, (Trusted ORACLE 7.0 and INFORMIX On-Line/Secure 5.0), against the TCSEC.

A method of analysis will be presented which is based on mapping decomposed TCSEC criteria (and interpretations to the Criteria) to the database through a detailed analysis of each product's documentation and users' manuals. This method could be applied to assist in the analysis of any DBMS software product. It should be noted, that this

1. Other countries, such as Canada and the European community have their own criteria for measuring the security effectiveness of computer systems.

comparative analysis is not a replacement for the official product evaluations conducted by the U.S. Government. The type of analysis described in this thesis would normally be done prior to the release of the official evaluation results published by the U.S. Government.

A. WHY IS THERE A NEED FOR THIS TYPE OF ANALYSIS?

The U.S. Department of Defense, through its subordinate agencies, the National Security Agency (NSA) and the National Computer Security Center (NCSC), conducts security evaluations of computer-related products. These products vary from computer operating systems to security add-on software packages, and DBMS software. The strenuous evaluation process often entails the development of both design and implementation evidence that the product under evaluation meets the requirements of a given trust level found within the TCSEC. Products are often available commercially, well before the evaluation by the NSA is complete.

Such is the case with the two DBMS products we have selected for a comparative analysis.[2] Our comparative analysis is an informal mapping of selected TCSEC requirements to the DBMS implementations (as described in the DBMS documents and manuals). In contrast, the NSA evaluations are conducted with the full cooperation of the vendors and allows for security testing techniques, source code inspection, and the review of proprietary information.

B. WHAT IS SECURITY?

Computer security is defined by some to be the secrecy of data (the prevention of unauthorized disclosure), the integrity of data (the prevention of unauthorized modification of data), and the prevention of denial of service (data is always available to the authorized user). [PFLE89] [GASS88] Therefore, the goals of computer security give the authorized computer user the assurance that their information is secret from others (within the system),

2. At the time of this research and writing the Final Evaluation Reports were not available for either Trusted ORACLE 7.0 or Informix On-Line/Secure 5.0. However, both products had been commercially available for almost two years and one year, respectively.

protected from modification by others, and that the information or computer resources are always available to them. Simply put, computer security focuses on secrecy, integrity, and denial of service.

The Privacy Act of 1974 and the Computer Security Act of 1987, which mandated that information be protected in automated information systems, is a driving force for both government and private industry to secure the data contained within there computer systems.

1. Security Objectives

The following sections further define the notion of what computer security is by additionally defining the three components: secrecy, integrity, and denial of service.

a. Secrecy

The secrecy component of security has been a prime focus of U.S. Government funded programs since the early 1970's. [GASS88] The objective is to protect the secrecy of classified information and government secrets. Because of the U.S. government's profound interest in secrecy, this aspect of computer security is well researched and studied by computer scientists [AMOR94]. Secrecy is intended to prevent the "leakage" of information from authorized users to unauthorized users.

b. Integrity

The integrity component of security covers the unauthorized modification of information stored in computer systems. Only authorized users of the system with the proper access to information should be able to alter (i.e., write, delete, append) data within the computer. If any other users change the information, then an integrity violation has occurred.(Note that authorized users of the system can still make erroneous changes to information and this would not be a integrity violation.)

Integrity of data has been a primary issue in the commercial business environment, with secrecy taking a secondary role [GASS88][AMOR94]. Businesses were

concerned mainly with the preservation of information used in their daily functions, because for them correct information or data saves time and money; its secrecy was less important. Recently, more companies have demonstrated an increased concern for secrecy.

c. *Denial of Service*

Denial of service is the least researched component of security and perhaps the hardest to implement and prove correct [GASS88][AMOR94]. The denial of service component of security involves the availability of computer assets by authorized users. Authorized users should always have access to information to which they are authorized; unauthorized users or other authorized users should not be able to intentionally deny access to information an authorized user has the authority to obtain.

Common forms of denial of service include things such as printers (or other devices) not available, or processors tied up because a job, with a higher priority, is running for an extended time period. Denial of service and integrity will not be a central focus of our comparative analysis.

2. Security Mechanisms

The components of computer security are addressed within a particular implementation of the system. Security can be addressed in all the layers of a computer system: hardware, operating system, and the DBMS.

a. *Hardware Security*

The first layer to provide security mechanisms in a secure computing system is found at its lowest level: the hardware. Hardware security mechanisms are usually the most primitive, and most easily verifiable security features of a computer system. Hardware can be, and is routinely validated to ensure that it is correctly implemented. [GASS88] (However, hardware still contains "bugs" just as software does, only less frequently). Once appropriately designed and developed, the security features within the hardware permit higher performance than comparable software and yield a

cleaner more reliable architecture.[GASS88] Hardware security is included in a computer system's evaluation, and may only be a peripheral consideration during a DBMS evaluation (or other application evaluation). The hardware and the operating system provide the "platform" on which the DBMS, (as well as other application programs) operate on.

b. Operating System Security

The second layer of security in a secure computing system is found at the operating system (OS) level. Access to objects (i.e., information containers having labels) is a primary focus of OS security. Discretionary and nondiscretionary access controls are present in most secure operating systems. Some operating systems have been developed which utilize a security kernel, which gives high assurance that a particular security policy is enforced. Many operating system products have been evaluated by the NSA since 1982 (the year evaluations began) [CHOK92]. Due to the system architecture of current DBMS implementations, certain security features of the OS (discretionary access controls and mandatory access controls on files and directories, for example) have to be explored when analyzing a DBMS product.

c. Database Security

A database management system (DBMS) is a complex software system designed to manipulate, store, and "manage" large amounts of raw data. Today's DBMSs are complicated, consisting of tens to hundreds of thousands of lines of code. Part of their specification requirements mandate them to provide for security and integrity of data. These security features are in addition to the security features provided by the underlying operating system.

The chief security features of standard (i.e., untrusted) database systems are account creation, account privileges, stored procedures and views, and audit logs. The more sophisticated DBMS products provide these and other more sophisticated mechanisms, while some of the lower-end products (such as PC based DBMS products) provide little or no security features at all.

Only within the last three years have commercially available DBMS products entered the evaluation process of the NSA. The two products we are analyzing, INFORMIX's On-Line/Secure 5.0 and ORACLE's Trusted ORACLE 7.0 will be the first two DBMS products to complete official evaluation by the NSA.

C. RESEARCH QUESTIONS

The primary focus of this thesis research is to compare and contrast the mandatory access controls (MAC) of two multilevel secure (MLS) DBMS products: INFORMIX's On-Line/Secure 5.0 and ORACLE's Trusted ORACLE 7.0. This includes a detailed look at which objects and subjects visible at the DBMS interface (e.g., schema objects) are labeled, and how the operating system mandatory access controls compare to the DBMS MAC functions. This thesis will focus on the following questions:

- How does each product interface support the MAC requirements, as identified in the TCSEC?
- How are labels structured, created, altered, and deleted by functions at each product interface, and do these implementations coincide with the requirements of the TCSEC?
- How do trusted subjects (i.e., MAC privileges) reveal themselves at the DBMS interface and what trusted subjects does each product support?
- Given both DBMS products, which implementation contains security features which exceed the B1 assurance requirements (as defined in the TCSEC requirements)?

D. ORGANIZATION OF THESIS

In order to establish the foundation for our security analysis of these two DBMS products, we present in Chapter II the trusted computer system concepts necessary to facilitate a discussion of the security features we are analyzing. Also, because the TCSEC is fundamental to our analysis, we have devoted Chapter III to a discussion of the TCSEC and its interpretations. Chapter IV is devoted to the operating system, HP-UX BLS 8.0,

which underlies both DBMS's. Chapters V and VI discuss the general architecture and operations of Trusted ORACLE and On-Line/Secure, respectively. The method of security analysis is presented in Chapter VII, and the detailed analysis for Trusted ORACLE and Informix-OnLine/Secure are presented in Chapters VIII and IX, respectively. A comparative analysis is presented in Chapter X, and conclusions, recommendations and future research are addressed in Chapter XI.

E. SCOPE, LIMITATIONS, AND ASSUMPTIONS

All information for this thesis was gathered from the open literature, interviews, and marketing documents. No special agreements have been arranged with the vendors or the evaluators for the disclosure of information about their respective software products.

II. TRUSTED COMPUTER SYSTEM CONCEPTS

A trusted computer system is one which has been evaluated by an evaluation entity and has received an assurance rating that the system will sufficiently enforce certain information security and integrity requirements. Proprietary and/or sensitive information can now be processed without an unacceptably high risk of compromise. The official evaluation entity within the United States (U.S.) is the NSA's National Computer Security Center (NCSC). The NCSC's *Trusted Computer System Evaluation Criteria* (TCSEC) defines a trusted computer system as:

> a system that employs sufficient hardware and software integrity measures to allow its use for the processing simultaneously a range of sensitive or classified information.[DOD85]

At the heart of the TCSEC definition is the notion of a reference monitor which controls a user's access to the sensitive information within the system. Sensitivity labels, such as "TOP SECRET" or "UNCLASSIFIED" are attached to the information and to all users' accounts (or programs) in the system. The problem with these differently labeled pieces of information and users is ensuring that the information will not be compromised. The challenge becomes one of controlling which users can access which pieces of information.

This issue of sharing multilevel data is briefly discussed in the next section and the reference monitor concept is discussed in detail in the section following. The remainder of this chapter discusses the basic concepts associated with trusted systems in general and is necessary for a discussion of the TCSEC in Chapter III.

A. MULTILEVEL CONTROLLED SHARING

A chief technical issue of trusted computer systems involves the multilevel sharing of a common computer system between many users and data objects (i.e., files, directories,

tables, etc.) at different classifications. A system which processes data at many different levels and whose users are also classified at many different levels is called a multilevel system (MLS). When a computer system must be responsible for access mediation, an evaluated system provides a level of assurance that access control policy is correctly enforced.

B. THE REFERENCE MONITOR CONCEPT

To address the multilevel sharing issue a new way of doing business was needed, so in the late 1960's serious work began to address this problem. Early experiences with computer security were characterized by "Tiger teams"[3] which went out and tried to penetrate a computer system's defense. Once the team penetrated the system controls and "broke into" the computer, another team of computer scientists would fix the holes discovered by the Tiger teams. This early method of computer security has been referred to as the "penetrate and patch" approach; systems were tested to uncover flaws, and the penetration paths uncovered were then patched.[NCSC92a] This process of discovering problems led to even more problems and soon a system became heavily patched with "spaghetti code" intended to prevent unauthorized users from entering the system without permission. This "penetrate and patch" methodology is unreliable because no one can decide when any more flaws exist. This was no way to establish that a system was secure. A more general approach was needed.

A research project performed on behalf of the DOD [ANDE72] produced the reference monitor concept in the early 1970's. In this concept of "a reference monitor which enforces the authorized access relationships between subjects and objects of a system" [DOD85], a fundamentally new approach to the multilevel sharing issue was formulated. The Anderson Report [ANDE72] described the architectural framework needed for dealing with the mediation of access in the face of potentially hostile users. [NCSC92a]

---

3. Tiger teams are teams of computer scientists who simulate adversaries or threats, and try to penetrate the security holes in a computer system. Primarily used in the DOD.

From this report, the notion of a reference validation mechanism (RVM) was described as "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."[DOD85] The report then listed the three design requirements that must be met by a reference validation mechanism [DOD85]:

- The reference validation mechanism must be tamper-proof.
- The reference validation mechanism must always be invoked.
- The reference validation mechanism must be small enough to be subject to analysis and tests; the completeness of which can be assured.

The reference monitor concept is depicted in Figure 1. This figure shows that the entities of interest, as mentioned earlier, are the "subjects" and "objects". A subject is an active entity created by the system, generally on behalf of a user (person). An object is a passive entity, usually a container which holds some type of information, such as a file. The reference monitor is an abstract machine that mediates the access of subjects to objects. When a subject tries to gain access to an object, the reference monitor determines if the access should be granted. This is done by applying a set of access control rules, found in a authorization database, and the sensitivity labels of subjects and the objects.

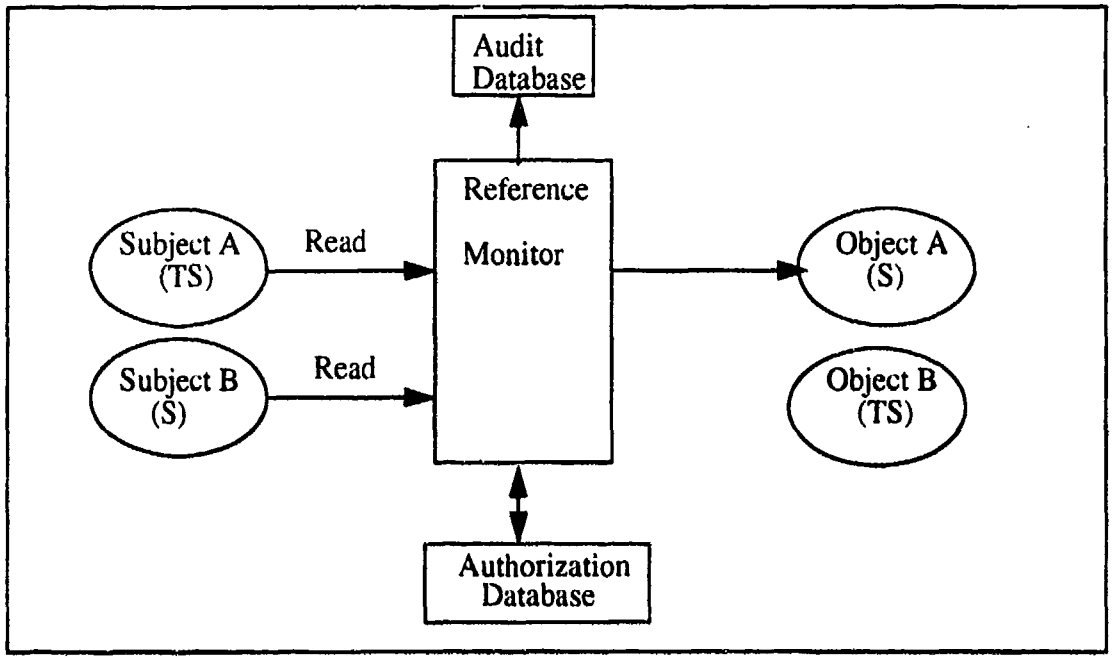


Figure 1: The Reference Monitor Concept

In Figure 1, Subject A is given permission to access (read) Object A. This is because Subject A's label (TOP SECRET) dominates (is greater than) Object A's label (SECRET). However, in the case of Subject B, its label (SECRET) is dominated by Object B's label (TOP SECRET) and is denied permission to access. A complete audit trail may be kept on all access attempts by writing to an audit database or audit file.

The reference monitor has become the general solution to the multilevel sharing problem. It is the most often used approach for building secure operating systems [GASS88] and represents an ideal approach for building access control features within trusted DBMSs. Before a discussion on how the reference monitor concept is implemented to control access to data and to resources (objects) by users (both authorized and unauthorized), the notion of a subject and an object is presented in the next section.

C. SUBJECTS

What exactly are subjects? A subject can be defined as "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." [DOD85]

A process is a "program in execution", a piece of code which is running on the processor. In a single-processor environment (i.e., only one central processing unit/CPU), only one process is active or running at any one moment. When the CPU is done with the process running, it switches to another process, which has been held in a ready state waiting for the CPU to take it.

Persons who use a computer system are represented by a unique "user process." Thus, the person using the system is really not the subject, but is only represented as such through the mapping of his user process back to himself. This process is created when the user logs on the system, and, upon successful authentication by the operating system (or the appropriate subsystem), is tagged with a unique identification (ID) string. This unique ID allows the operating system to map this login user process to one and to only one real world person.

A process is considered an active entity within the system, at which point we can define it as a subject. Likewise, any process which can access (read or write) an object is considered a subject. Later, when other commands (such as *ls* for list directory in UNIX) are initiated, new processes are spawned or forked. These new processes operate on behalf of and as a surrogates for the original user logon process. All the surrogate processes inherit the unique ID of the user process which invoked it, but in some instances users may invoke subjects possessing another user's unique ID (i.e., the *su* or superuser command in UNIX). [GASS88]

D. OBJECTS

The previous section discussed the subjects within the computer system. This section will discuss what an object is, the different categories of objects, operating system and

database objects, and object identification. This area is especially important in the area of DBMS evaluations and has been the topic of some discussion in the context of DBMS objects versus operating system objects [GRAU90].

An object is defined as 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, programs, bits, bytes, words, fields, processors, video displays, keyboards, clocks, printers, and network nodes."[DOD85]

An object can be thought of as a container, like a bucket, which can hold data. This bucket can be filled up (written to in the computer vernacular) and/or drained (read from) by the active processes (subjects, as previously discussed) in the system. However, this is a special bucket, when you drain it, the data inside does not really move at all, so no data is ever lost when the bucket is drained. In the technical sense, this object is a repository of data, which has an internal state that is changed (written) and/or observed (read) by the subjects of the computer system [NCSC89]. All state changes of the object are initiated by a set of well-defined operations that are available to the subjects[NCSC89]. One could call this category of objects "data objects", or one could categorize them as storage objects or named objects, depending upon how they are created and managed by the TCB. (See section below.)

1. Object Categories

Data objects can be thought of as containers which hold data and is a broad category in which to classify objects. One way to classify objects is by their physical properties, such as memory blocks or segments. Another useful way that we can classify objects is by the way in which subjects can access them [NCSC89]. In the context of DBMSs, if subjects can access objects through discretionary access controls, these objects are called "named objects." If subjects can access objects through mandatory access controls, then these objects are called "storage objects." [NCSC89]

2. Operating System Objects

Generally, any passive entity which contains data of any sort and can be manipulated by an operating system subject (active process) can be considered to be an operating system object. These include files, directories, special files, interprocess communication (IPC) objects, pipes, and symbolic links.

3. Database Objects

The objects within a DBMS are often different from the objects within an operating system environment. DBMS objects are usually defined with a finer granularity than those within an OS. For instance, an OS data file may contain several database tables. The OS recognizes only the OS objects (e.g., the files), whereas the DBMS recognizes the tables within the file.

Objects found in most DBMSs include tables, views, indexes, and clusters. The database is also considered an object (because it too is a large container of information), as are the rows within the tables. (In some research databases, the attributes and elements (i.e., tuples) within the rows are also considered objects). In the later chapters on the ORACLE and INFORMIX architectures and operations, we will discuss the specific objects found within those systems. (See Chapter V and Chapter VI, respectively).

E. PRIVILEGE

The notion of privilege is an important security consideration when designing a security policy for a computer system (See "SECURITY POLICY" on page 18.) and when implementing that security policy into a security mechanism for protecting the system (See "SECURITY IMPLEMENTATIONS" on page 26.). A privilege is a right given to a process or subject, so that the process can perform certain functions. These rights may permit the process to access only certain resources within the system, such as only certain memory spaces or registers, or only certain I/O devices, such as selected printers and disk drives. When a process is given access to all the resources in the system, it is considered to

be in a "privileged mode". A process operating in a privileged mode can access more of the memory address space (i.e., more objects), or it can invoke special system functions.

For example, the person who operates and maintains the computer system is called the system administrator. The system administrator usually requires special privileges to keep the computer system running correctly.

Another example of a special privilege is a system's backup and recovery program. The backup program will be allowed to bypass read restrictions on files so it can copy them to a magnetic tape or a floppy disk; the recovery program will be allowed to bypass write restrictions on files so it can write files and restore them to original form.[GASS88]

1. Least Privilege

This notion of different privileges given to the processes running in the system was inspired by the principle of least privilege. Least privilege is the concept that users (subjects)be granted only as many resources as they need to complete their job. The Orange Book defines least privilege explicitly as:

> ...that each subject in a system be granted the most restrictive set of privileges (or lowest clearance) needed for the performance of authorized tasks. "The application of this principle limits the damage that can result from accident, error, or unauthorized use.[DOD85]

The idea is to reduce the number of potential interactions between programs to the minimum amount needed for correct operation, so that if erroneous input is introduced by users or improper functions are called, the amount of damage to the system will be minimized. [SALT75]

2. Modes of Execution

The concept of modes of execution is crucial for the enforcement of least privilege. A mode of execution (or domain of execution) is the environment in which a process (subject) operates, and contains all those resources (objects) for which it has access. The Intel's iAPX x86 CPU, can operate in four modes. Assuming the programmers

programming in the system are aware of this functionality, they can write programs which operate in the different modes, thus protecting one program from another. (The UNIX operating system is written to take advantage of two modes, the *system* and the *user* mode). Any architecture with two or more states provides a finer degree of control, because each mode is ordered from most privileged to least privilege. The less privileged mode has access to less memory space than the previous (higher privileged) mode. Other names for modes include domains, states, rings, or context. [GASS88]

In Figure 2, the concept of modes or domains is demonstrated with rings. In the figure, the most privileged domain is the innermost domain (0) where for example the hardware operates. As one travels outwards, the less privileged rings are encountered (rings 1-4). The security kernel (the part of the operating system that does the security checking) and the operating system are usually the most privileged software in the computing environment: therefore, they are shown as inner rings (1 and 2). The DBMS (shown as ring 3) resides on top of the operating system and the hardware. The processes running in the DBMS are not allowed access to the more privileged operating system code except through certain "gates" (i.e., system calls) into those inner rings. In other words, it can only look outwards and access the code of the users applications which reside on top of it in ring 4.

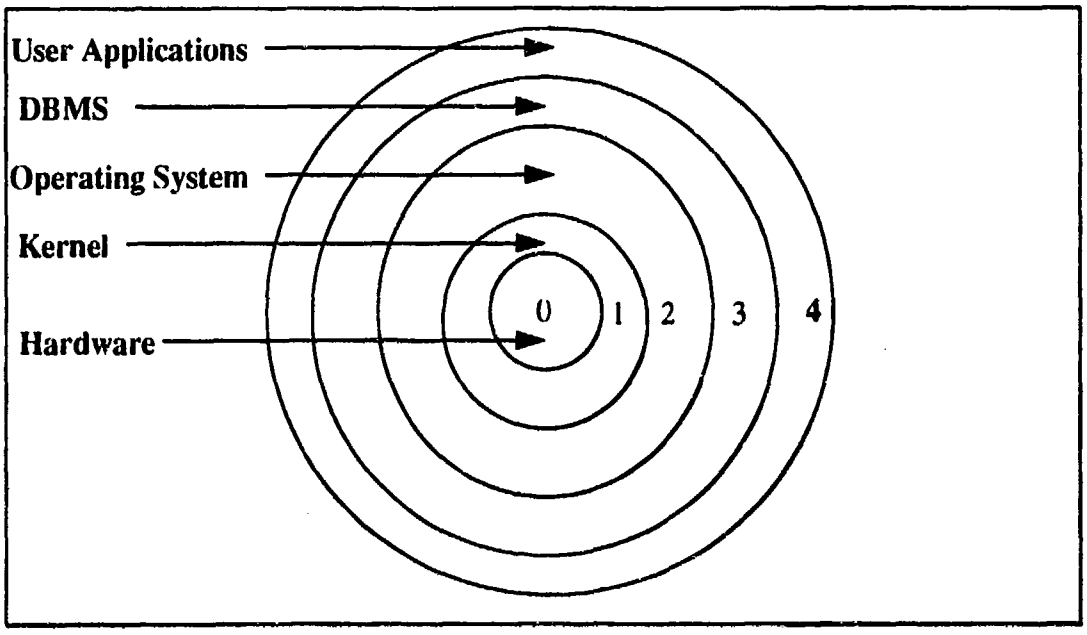


Figure 2: Hierarchial Domains or Rings

When a process executes, its respective subjects (a process may have more than one subject) operate within a predetermined ring or domain. A domain represents all the objects to which the subject has access (read or write). A subject's domain at any particular time might include a variety of programs, files, data segments, and I/O devices such as printers and terminals. Such a domain is shown in Figure 3.

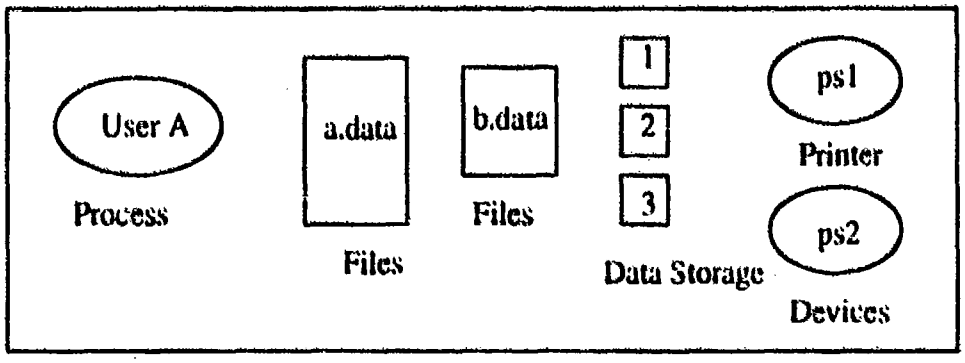


Figure 3: Domain of Execution

As the user "A" subject executes, in Figure 3 above, it is restricted to those objects within its domain of execution. This process can access both data files a and b, data storage segments 1, 2, and 3, and the I/O devices ps1 and ps2. This is all the subject can access in this domain. The process must change domains to access more objects. It does this if enabled with some privilege, which allows it to transfer to a more privileged ring or domain.

The important security concept here is that the mode of operation protects each ring from the outside and allows each ring to control programs efficiently in the less privileged rings. [GASS88] Least privilege is realized by allowing a process to only operate within a ring or mode that has just the necessary number of objects which it needs to complete its job.

3. Separation of Privilege

One last important aspect of privilege is the need to separate privileges from users so that the ability to perform certain functions are distributed among different users. For example, it may not be prudent in some environments to give a system administrater all the functions needed to operate and maintain the system. Perhaps the function of creating and setting up new users on the system should be handled by a second person, and the ability to acquire audit records and audit trails by a third person. This way the system administrator can not create bogus users (perhaps to run malicious programs to steal information) and then try to hide or destroy the audit records. One breach of trust by a user with all privileges could compromise the entire system and destroy valuable information which cannot be retrieved.

F. SECURITY POLICY

The builders of a high security computer system must state the desired requirements for security before building the system, if they have any hope of realizing security in the end product. These computer architects start with a security policy. A security policy can be defined as:

the set of rules, directives, and practices that regulate how an organization manages, protects, and distributes sensitive information.[DOD85]

The security policy describes every aspect of how information will be handled both inside the system and outside the system. Sensitive information is defined and could include everything from government TOP SECRET documents to a small company's proprietary business information or its personnel database. The security policy, once defined, is then translated into a security implementation within the computer system. The desired attributes of the system are eventually realized, in part, by the implementation of some specific set of mechanisms; functions which can be shown to provide the required attributes. [NCSC92a] The critical point is that one starts with a security policy (i.e., a high-level statement of the desired global properties or characteristics of the system), then proceeds through a number of refinement steps culminating with a set of specific implementations. [NCSC92a] See "SECURITY IMPLEMENTATIONS" on page 26.

1. Security Models

The security policy of a computing system can vary from short (i.e., no person outside the company should access this data) to extremely complex (e.g., U.S. Government TOP SECRET information systems). Formal security policies have been proposed and formulated into security models for several years. Both single-level and multi-level security models exist, but for the purposes of this thesis, only multi-level security models are of interest. In the multi-level world, multi-level security models, where many different objects and subjects of different classification are present, is our focus.

a. Military Security Model

We start with the military security model because most secure computer systems used by the defense and national security establishments are based on this model. This model also represents the base upon which many other important multilevel security models are built (i.e., the Bell-LaPadula Model).

Within the military security model all subjects and objects are labeled as UNCLASSIFIED, CONFIDENTIAL, SECRET, or TOP SECRET. We call these labels security levels. The rankings of these pieces of information are mutually exclusive (disjoint), and describe the sensitivity of the information. These sensitivity labels represent an implementation of a more general model called a lattice. All elements are ordered under a partial ordering of "dominates." (One level is greater than or equal to another.) This lattice forms a linear hierarchy where elements, if comparable, have a definite order in the model. All elements can be determined to be > or <= to any other element. The military lattice is shown in Figure 4.

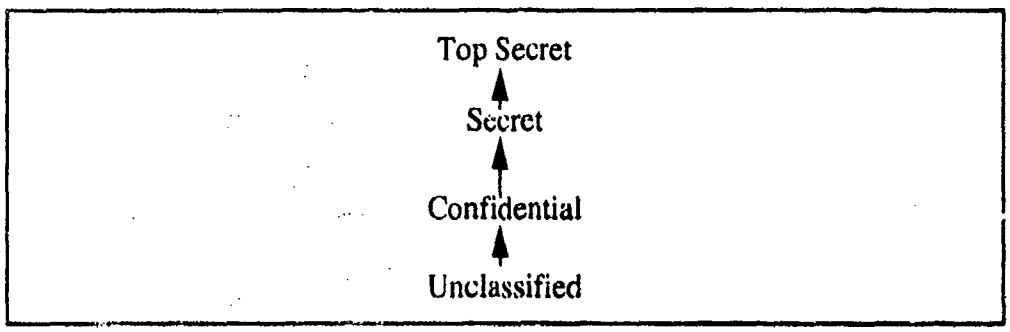


Figure 4: Military Lattice

The military security model utilizes two important principles are utilized: least privilege and need-to-know. The principle of least privilege was discussed previously and as applied here states that a person requiring access to SECRET information to do his/her job, get only SECRET (or anything less than SECRET) information, and not TOP SECRET information. There is no need to give the user more information then they need to do their work; in other words give the user the least-privileged information (i.e., SECRET) needed to accomplish their mission.

The second principle is the "need-to-know" rule; access to sensitive information is allowed only if the user is approved for the category in which the information is being sought. Each piece of information, in addition to the security label, is associated with zero or more compartments, describing the subject matter of the

information. These compartments are then used to enforce the need-to-know principle, so that users can obtain access only to information which is relevant to their jobs. Examples of compartments are TERRORISTS, NUCLEAR, and SPIES. A single piece of information would then be coded with zero or more compartments, depending on the categories to which the information applies (see Figure 5). For a person to gain access to a piece of information, he/she must possess all the compartments associated with the information, as well as a sensitivity classification that dominates the label of the information. For example, if Captain Smith wants TOP SECRET information that deals with nuclear spies and terrorists, he must have a security rating of at least TOP SECRET and compartment clearances for NUCLEAR, SPIES, and TERRORISTS. (See row 1 of Figure 5) Clearances in rows 2-5 (Figure 5) would not be adequate to gain the desired information.

1. TOP SECRET:NUCLEAR,SPIES,TERRORISTS
2. SECRET: SPIES
3. SECRET: TERRORISTS
4. CONFIDENTIAL: NUCLEAR
5. UNCLASSIFIED:

Figure 5: Military Security Labels

b. Bell-LaPadula Model

One of the best known and most popular multilevel security models is the Bell and LaPadula model developed and published by D. Bell and L. LaPadula in 1973. [BELL73] The Bell-LaPadula model (BLP) describes the allowable paths of access control in a secure system. The goal of the model is to identify allowable communication channels where it is important to maintain secrecy. (Note that this model does not preserve the integrity of the information).The model has been used to define the security requirements for systems concurrently handling data at different sensitivity levels. [PFLE89] This model

was created to obey the military security model and is sometimes referred to as a multilevel security model. [GASS88]

The BLP model has been widely accepted as the model of design when building multilevel secure systems [GASS88]. The reason is that it represents one of the best available models for describing the acceptable connections between subjects and objects of different levels of sensitivity. With the BLP model, a machine can be built that can process data of two or more different sensitivity levels.

Two properties characterize the Bell-LaPadula Model as noted below:

- Simple Security Property. A subject can only read an object if the security label of the subject dominates (greater than or equal to) the security label of the object.
- *-Property (or Confinement Property). A subject can write (modify) an object only if the subject's security label is dominated (less than or equal to) by the object's security label.

The implications of these two properties are shown in Figure 6 from [PFLE89]. The classification of subjects (represented by squares) and objects (represented by circles) are indicated by their positions; subjects and objects higher in the figure represent higher levels of sensitivity.

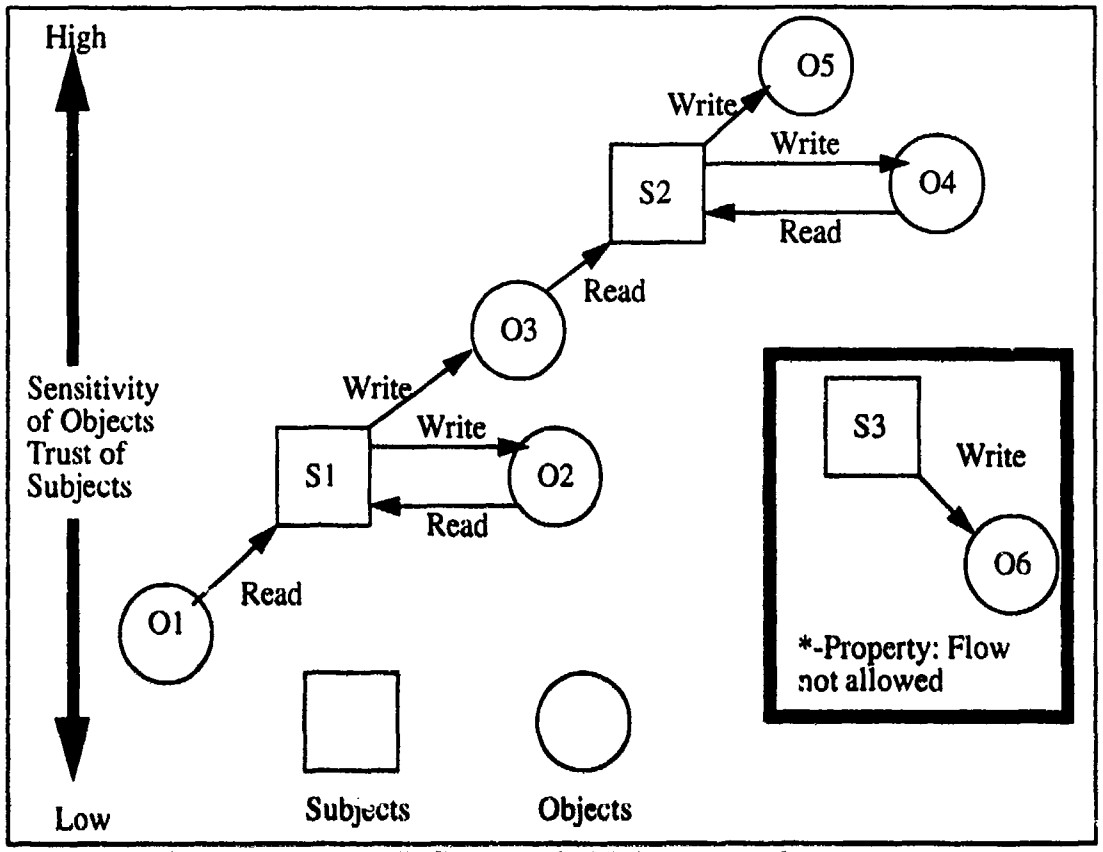


Figure 6: The Bell-LaPadula Model demonstrating the secure flow of information [PFLE89]

The simple security property means that a subject can read objects at its security classification and below. For example, a SECRET subject can read SECRET, CONFIDENTIAL, and UNCLASSIFIED data.

The *-property of the Bell-LaPadula model is used to prevent write-down, so that a subject with a high classification (which has access to high-level information objects) cannot copy information into a lower level object. (See flow demonstrated in Figure 6 in heavy black box; this is not allowed.) This is synonymous with the military security model, which prevents persons with TOP SECRET clearances to give TOP SECRET information away to UNCLASSIFIED users.

2. Access Control Policies

The following sections briefly defined some of the important access control policies or rules found in secure systems.

a. Supporting Policy

Obtaining access into a computer system, usually the process of logging on the computer terminal, is handled by the identification and authentication (I&A) subsystem of the operating system which runs the computer. Even though this I&A subsystem is normally not considered part of the access control policy, it will be discussed here because it is an overall part of the security policy.

The identification part of the I&A subsystem is used by the operating system to identify the user who logs into the computer system. This is usually done by the user typing his/her user's name (a predefined username ID) at the logon prompt. This identification string is unique to the system and allows the system to identify an individual user.

The authentication part of the I&A subsystem is used by the operating system to ensure that the identification presented to the system is in fact the real user who was assigned that username ID. This is usually done by prompting the user to enter a password that only he/she knows. If the password is entered correctly, the system accepts the identification as genuine. Now any auditing enforced by the operating system (or other applications) can correctly identify not only the username ID that performed an operation but the actual user (person) that performed the actions.

b. Discretionary Access Control

Discretionary access control became a serious issue with the emergence of multiuser systems and the sharing of files stored on mass storage [GASS88]. Controlling access to disk files was probably the first widespread computer security concern, because for the first time the system, rather than the operator, was required to enforce access control [GASS88].

Discretionary access control permits the owner of an object, such as a file, to authorize access to that object by other subjects in the system. This creator, at his/her own discretion, determines who is authorized to access the objects he/she creates. Discretionary access control is best demonstrated by way of an access control matrix.

c. Access Control Matrix

The access control matrix is a table in which each row represents a subject, each column represents an object, and each entry is the set of access rights for that subject to that object. [PFLE89] This matrix may in fact be sparse because not every row and column intersection will have an entry; most subjects will not have access rights to most objects. In Table 1 below, an access control matrix is shown; objects are shown in the double-boxed columns and subjects represent rows. The allowable modes of operation (i.e., rights) for each subject are: o (owner), r (read), w (write), x (execute). (Note that some boxes are blank).

TABLE 1: ACCESS CONTROL MATRIX

Subject	file a	file b	file c	help.t	compiler	linker	clock	printer
USER A	orw	orw	orw	r	x	x	r	w
USER B	r			r	x	x	r	w
USER C	rw			r	r	x	r	w
USER D			r	r	x	x	r	w
sys mgr	-	-	-	rw	ox	ox	orw	o

Because the access matrix can be represented as a list of triples (subject, object, rights), searching a large number of triples can be time consuming and inefficient. Therefore, the access control matrix is used more as an abstraction than a real implementations.

d. Mandatory Access Control

In a variety of situations, discretionary access control may not be acceptable policy. Therefore, mandatory access controls are imposed which cannot be bypassed, even indirectly, by the subjects within the system.

Under mandatory access control (MAC), subjects and objects are assigned special security attributes, usually security labels, which are tranquil and cannot be changed. (This is opposite of the DAC policy which allows users (creators of objects) to change the attributes.) The system reference validation mechanism determines if a subject can access a particular object by comparing the security label of the subject and the security label of the object (and the DAC attributes).

Mandatory controls are used in conjunction with discretionary access controls and serve as an additional restriction on access to objects. A subject may have access to an object only if the subject passes both discretionary and mandatory access control checks. [GASS88] Since users cannot directly manipulate mandatory access control attributes, users employ discretionary controls for their own protection from other users. [GASS88] Mandatory access controls come into play automatically as a stronger level of protection that cannot be bypassed by users through accidental or intentional misuse of discretionary controls. [GASS88]

G. SECURITY IMPLEMENTATIONS

Thus far, the discussion has traversed from a security policy, which can be represented by a security model, to the access control policies (or rules) which are made up of the DAC and MAC elements. Their are several popular security mechanisms that realize the implementation of these rules and policies.

1. Protected Groups or Directories

A very simple access control mechanism is a file directory which controls the access to the set of files within the directory by a set of subjects within the computer system. Each file within the directory has an owner who possesses major access rights, including

the rights to declare who has what access and to revoke access by any person whenever desired. Each user has a file directory, which lists all the files to which that user has access.

The UNIX operating system's file ownership and permissions are a good example of this directory type implementation, and as discussed earlier, of discretionary access control. UNIX implements a very simple mechanism which uses only a few bits of access control information attached to each file. Every file has an owner who created that file. The files created by the owner are so listed with that ownership, and file permissions, or modes, are associated with that particular file. A total of ten bits are used to indicate which permissions are applied to three different entities: user (owner), group, and others. Each entity can have permission to read (r), write (w), or execute (e) the file as a program. See Figure 7 below.

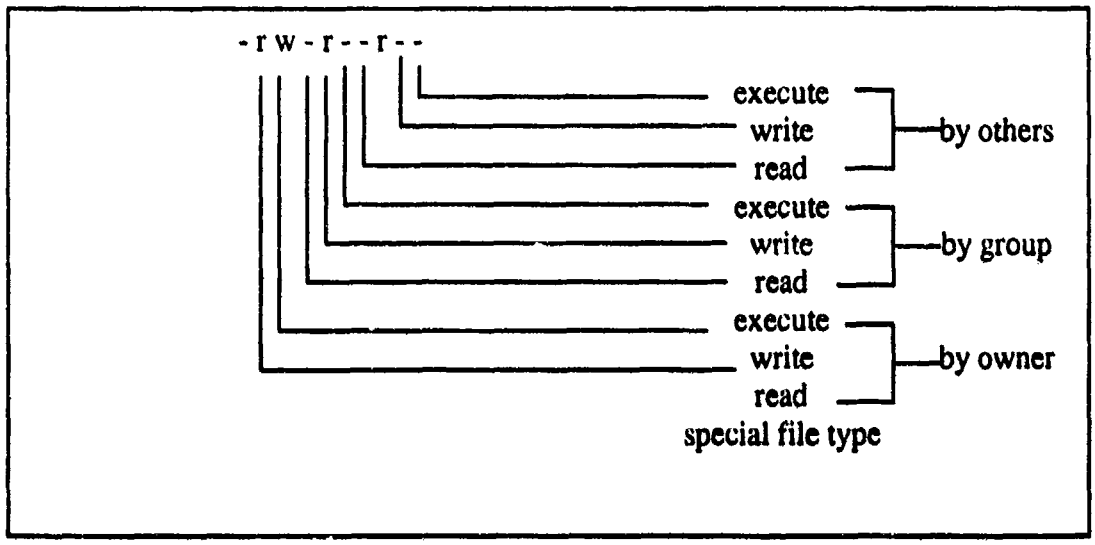


Figure 7: UNIX Protection System

A dash indicates that a permission is not enabled. The left most bit is used to indicate a special file type, like 'd' for directory. Typical files will have a dash in this location.

2. Access Control List System

Another access control mechanism and one of the most effective access control schemes is the access control list (ACL) [GASS88]. In this implementation every object

has one access control list associated with it. This list shows all the subjects who can have access to this object and what their access rights are (i.e., read or write). [PFLE89] The access control list identifies the individual users or groups of users who may access an object, such as a file. Because all the access control information for a file is stored in one place and is clearly associated with the file, identifying who has access to a file, and adding and deleting names to the list can be done very efficiently [GASS88].

An example of how an ACL works follows: if subjects *john* and *beth* both have access to a file *game*, the operating system will maintain just one list for the object (file *game)* showing the access rights for *john* and *beth*. The access control list can have general default entries (*, for wildcard) for any users which allows a public file or program to be shared by all possible users of the system. See Figure 8 below. File *public* represents an ACL which any user in the system can access the file. File *game* shows *john* and *beth* each with there own specific rights; read write or execute.

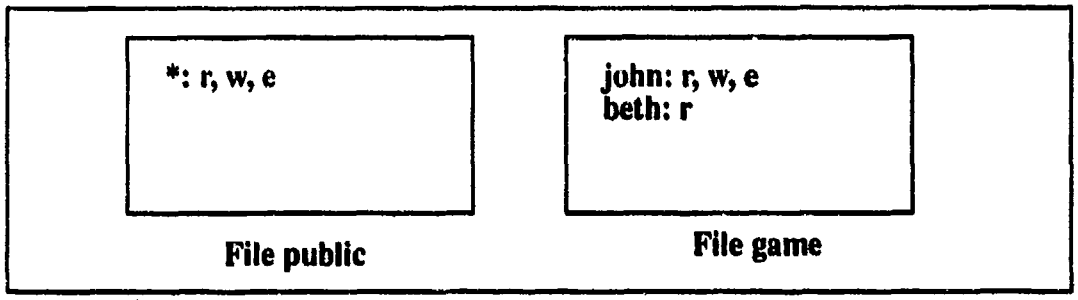


Figure 8: Access Control Lists

The access control list must be examined each time any user accesses an object. This can lead to poor performance when many objects need to be accessed because many checks must be made against the list. However, with suitable defaults and groupings of users, access control lists rarely require more than a handful of entries. [GASS88] Another disadvantage is storage management because the length of the access control list is not fixed, but instead variable. Maintaining a variable length list for each file results in either complex directory structure or wasted space for unused entries. [GASS88]

H. TRUSTED COMPUTING BASE

We have now traversed to the point where we must now place within the system the security mechanisms we have implemented (i.e., protected directories, ACLs, capabilities, etc.). The location where we placed the security mechanisms is inside a perimeter we call the trusted computing base. The trusted computing base (TCB) has been defined by the TCSEC as:

> 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. [DOD85]

The TCB contains all the necessary mechanisms needed to provide for the security of the computing system in accordance with the defined security policy. The incorrect operation of the mechanisms within the TCB could lead to the unauthorized disclosure of information or another security violation relative to the system's security policy.

To further define and develop this notion of the TCB, the boundaries of the system must be identified. Two boundaries are of importance as discussed in [GASS88] are the system boundary and the security boundary.

1. System Boundary

The computer system's boundary or interface with the outside world, must be clearly defined, and the threats from the outside world must be identified and evaluated before a security policy can be developed. The system is composed of all the computing hardware, firmware, and software, and includes all the telecommunication hardware and software as well (i.e., networks, phone-lines, wireless, etc.). Everything identified as being inside or part of the system, must be protected by the system. Everything outside the system is left unprotected by the system. [GASS88] The threats to the system must be made a primary focus during the security plan development. [GASS88]

To identify the system boundary, the interfaces between the computer system and the outside world must be specified.[GASS88] External controls, such as physical controls and personnel and procedural controls are setup to enforce this interface. Items which are outside this interface are the users, terminals, some I/O devices such as line printers (only some printers, not all), and data storage media, (e.g., archive tapes stored off-line). As long as the external controls enforce this interface properly, the threats from outside the boundary can be keep out.

Those items allowed inside the system boundaries, such as authorized users or programs, are monitored by the internal security controls. The internal controls are implemented within the hardware and software of the system, and their primary purpose is protection of information within the system against the specified threats. However, if unauthorized users or processes bypass the external controls of the system, the internal controls cannot be guaranteed to stop the threat and to protect the system[GASS88]. For example, if a system administrator gave his password to another user, this person would be able to get through the external controls (i.e., policy of not giving away passwords is violated) and manipulate the system anyway he/she is capable. The internal controls can in no way stop this user and protect the system. Once external security controls are broken, the system is vulnerable, and no amount of internal security controls can be expected to stop the intruder from harming the system.

2. Security Perimeter

According to [GASS88], the components inside the computer system can be classified into two types: those responsible for maintaining the security of the system, and all others. The boundary separating these two components is called the security perimeter. (Every thing inside the security perimeter is the TCB). Within this security perimeter lie that part of the operating system responsible for security (i.e., security kernel) and the hardware and firmware; outside this perimeter, but inside the system boundary, lie the user programs, data files, terminals, and I/O devices controlled by the system. The components

within the security perimeter must be precisely defined, because once they malfunction, a security violation can occur.

This interface, called the security boundary, must be well defined just like the system boundary interface. [GASS88] This interface is controlled and enforced by the system's security relevant components. In a system which utilizes a security kernel, the list of system calls between the security kernel and the operating system is a good example of the interface between the two components of the system. In Figure 9 below, the system's hardware, security kernel, a portion of the OS, and a portion of the DBMS comprise the TCB. (Builders of secure systems try to minimize the size of the TCB to make validation easier.) The users of the system are outside the system and the external controls prevent known threats from entering the system. The TCB is maintained by the security relevant components of the system and is responsible for all security decisions.

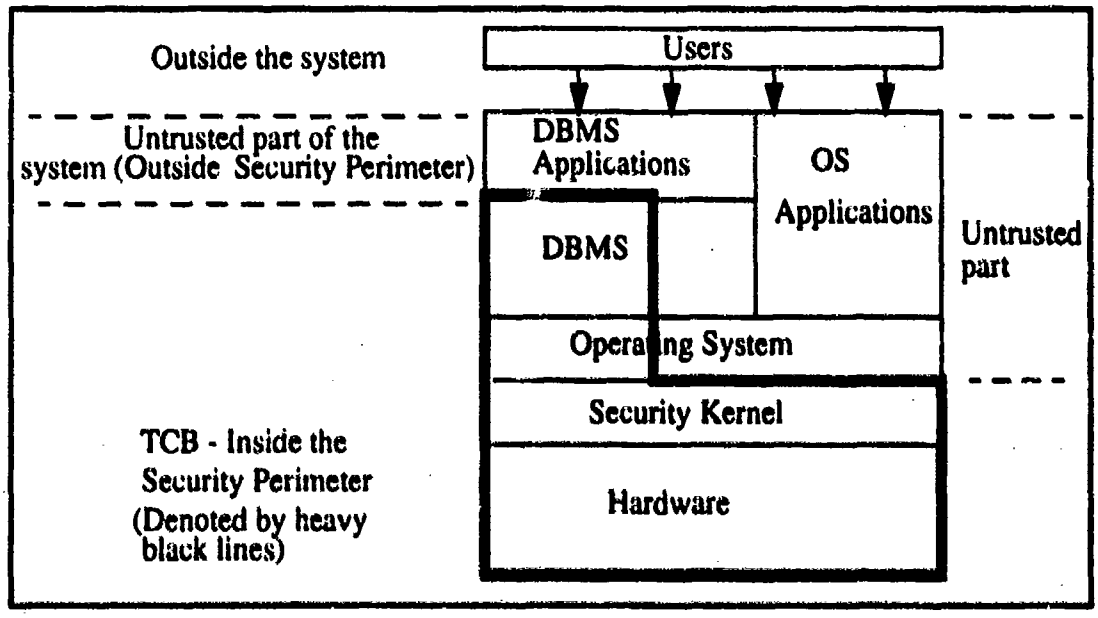


Figure 9: System Boundaries and the Trusted Computing Base

The TCB and the internal security controls, which comprise the TCB, are the primary focus of the design and evaluation of trusted systems.

I. TRUSTED SYSTEMS

Trusted systems are "trusted" only if they (their design documentation) provide convincing arguments or proofs that the security mechanisms work as advertised and cannot be subverted or disabled. [VETT90] This chapter has laid out some of the fundamental concepts behind the design of trusted systems. The remainder of this chapter presents the fundamentals for the evaluation of trusted systems. The following two sections deal with two fundamental topics in the evaluation of trusted products: TCB subsetting and trusted subjects.

1. TCB Subsetting

TCB subsetting is a design approach to building secure computer systems. It can be used initially when a system is first built or it can reuse and extend previously built and verified trusted systems.[VETT90] It is motivated by the need to be able to extend a TCB by building on an existing TCB (such as a security kernel or an operating system) without disturbing its basis for verification. This is essential when a vendor wants to build a trusted database system on another vendor's trusted operating system. [LUNT92] TCB subsetting allows an "evaluation by parts" which is a technique used to evaluate a software product or system in modules instead of all at once.

The TCB subsetting concept evolved from work by Schaefer and Schell [SCHA84] and Shockley and Schell [SHOC87] on extensible TCBs. This subsetting approach allows the TCB to be structured into layers, (and later decomposed) with each layer enforcing its own policies and with each layer constrained by the policies enforced by the layers beneath it. Ideally, the lowest layer is a mandatory TCB that enforces mandatory access control for all the layers above it. This is particularly useful when one vendor builds a layer of the TCB enforcing a discretionary security policy on another vendor's mandatory TCB. TCB subsetting not only allows reuse of existing TCBs, but also permits the evaluation process to take advantage of an already evaluated lower TCB. Figure 10 represents a typical TCB subset architecture.

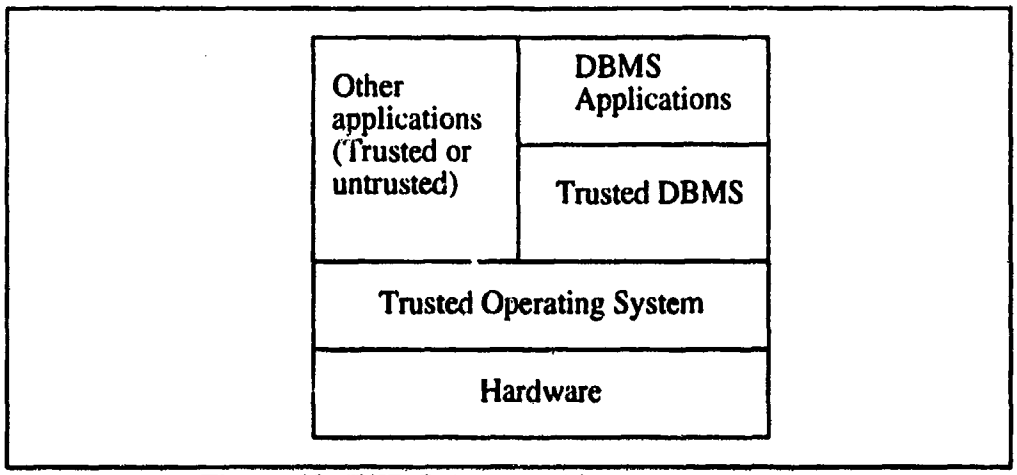


Figure 10: TCB Subset Architecture

An advantage of the TCB subsetting approach is that it allows vendors building an upper level TCB to take advantage of the security features provided from the lower TCB upon which they build their new product. For example, if a trusted operating system provides for mandatory security between subjects and objects, then the newly built DBMS need only enforce additional discretionary security needed by the database. Thus, building multilevel DBMSs using this approach may be the quickest and the most viable approach to getting a multilevel DBMS product evaluated. [LUNT92]

The use of TCB subsetting also can provide the greatest degree of security possible for mandatory security.[LUNT92] Because there is no trusted MAC component in the DBMS itself, the risk of disclosure of sensitive data is considerably reduced. This is because the DBMS is governed by the mandatory TCB of the underlying operating system, which partitions multilevel data by their classification. Thus the subjects within the DBMS, when operating on behalf of the users, cannot gain access to any data whose classification is not dominated by the users' clearance. This means that database operations can be handled by subjects which are single-level and untrusted with respect to mandatory access controls. This is the most conservative approach possible for mandatory security. [LUNT92]

2. Trusted Subjects

Among computer scientists and trusted system builders, there has probably been more controversy and misunderstanding about the concept of trusted subjects than about any other single aspect of secure systems. [GASS88] Trusted subjects are important to understand because most of the secure MLS DBMSs being evaluated today utilize a "trusted subject architecture."

The "trusted subject" term originated in the Bell and LaPadula security model.[GASS88] Trusted subjects are not required to obey the *-property (confinement property) of the Bell and Lapadula model. Trusted subjects are trusted not to violate the security policy of the system.

The trusted subject approach to designing and evaluating a DBMS is an alternative to the TCB subsetting approach discussed earlier. The trusted subject approach is implemented by making an upper level application, such as a DBMS, a trusted subject so that it will not violate the security policies of the underlying TCB.

For most of the multilevel secure DBMSs, the DBMS runs on top of a trusted operating system. However, since the trusted DBMS operates with certain policy enforcement mechanisms that allow it to enforce mandatory access controls (on its own objects, such as tables and rows), the DBMS must execute as a "trusted subject" with respect to the operating system TCB. [DOYL91] In other words, the DBMS now enforces some aspect of MAC, instead of the operating system enforcing it all. The DBMS is trusted that it will do it right and not violate the security policy.

Logically, trusted subjects are part of the TCB and use the services provided by the TCB, but architecturally they run as subjects in a domain outside the basic TCB. In a sense, trusted subjects are just the extensions of the TCB [GASS88]. When evaluating a product (or system) using the trusted subject concept, the more privileged TCB component must be combined into a single TCB subset with the trusted subject.

The use of the trusted subject approach often significantly reduces the total effort involved in evaluating the trusted system product, because it is only necessary to show that

the trusted subject does not permit unauthorized information flow, as opposed to showing that it correctly enforces an access control policy. [NCSC89]

One disadvantage of the trusted subject methodology is the definition of how the trusted subject (i.e., DBMS) is used, because its use can cause new information flows beyond those that can be discovered by performing a flow analysis exclusively of the trusted subject. Such flows can be discovered only by performing a flow analysis on the combination of the trusted subject and the underlying TCB (i.e., the operating system), a task which may in and of itself be very difficult to accomplish.

III. TCSEC

The previous chapter discussed the trusted system fundamentals absent of the specific requirements needed for a computer system to rate an assurance level as described in the DOD Trusted Computer System Evaluation Criteria (TCSEC).[DOD85] The TCSEC (also known as the "Orange Book" or the "Criteria") is the primary document outlining the criteria for the evaluation of trusted computer systems and products. This document specifically outlines the requirements of each assurance level or rating class. Thus far, the TCSEC is the U.S. standard on which all trusted products are evaluated.[4]

A. THE NEED FOR AN EVALUATION STANDARD

The U.S. Government has a legitimate interest in ensuring that the computing systems it acquires protect information with some level of assurance. The government is chiefly concerned with nondisclosure or secrecy of information. In addition, since no design and evaluation standard existed anywhere in the world in the 1970's, a common body of knowledge was needed to begin the process of evaluating computer systems for secure environments. The Criteria's initial purpose was focused almost exclusively on the acquisition of computer systems for the national security establishment. [DOD85]

The computer industry also required a standard by which to design and build secure computing systems. It was in the best interest of the U.S. Government and the NSA to encourage the development of trusted computer systems and products, thus making them widely available in the commercial market place. [DOD85] With a vast inventory of trusted systems and products available, the NSA, the DOD, and other interested government agencies could pick and choose the best product or system for the right situation or environment.

---

4. There is presently a move to implement an international standard called the Common Criteria. More will be said on this at the end of this chapter.

Over the past decade, as the Criteria have matured (i.e., new interpretations issued) and additional publications (i.e., from the Technical Guidelines Program) have emerged, more commercial developers have moved towards building secure computer systems. In the beginning, manufacturers built systems using the Criteria because it was mandated by the U.S. Government. However, now commercial and private companies have realized the need to better protect proprietary information, personnel databases, and other private or sensitive information. Because of this, the use of trusted systems is becoming more widespread outside the government establishment [GASS88].

B. HISTORY OF THE TCSEC

The National Computer Security Center (NCSC) is part of the National Security Agency (NSA), an agency of the U.S. Department of Defense (DOD). In January 1981, the Department of Defense assigned the responsibility for computer security to the Director of the National Security Agency (NSA). This action led to the formation of the Computer Security Center, whose charter was promulgated in the DOD Directive 5215.1 in October 1982. It specifically tasked the Computer Security Center to establish and maintain:

> technical standards and criteria for the security evaluation of trusted computer systems that can be incorporated readily into the Department of Defense component life-cycle management process.[NCSC90]

The NCSC, in conjunction with other components of the NSA (e.g., Information Systems Security Organization-ISSO), is involved in establishing computer security criteria and guidelines such as the TCSEC, evaluating computer hardware and software products for security and assurance against the Criteria, and conducting and supporting computer security research and development.

Before the Computer Security Center was established, two departments of the U.S. government were instrumental in the establishment of computer security standards and criteria for evaluating computer system products. They were the DOD and the Department of Commerce (DOC).

Work began as early as 1967 with a DOD task force organized to address computer security safeguards that would protect classified information in remote-access computer systems. [DOD85] The report from this task force (the RAND report) made a number of policy and technical recommendations on actions to be taken to reduce the threat of compromise of classified information processed on remote-access computer systems. [DOD85]

The DOD responded to the RAND report and issued DOD Directive 5200.28 (1972) and its accompanying manual DOD 5200.28-M (1973). These decrees established uniform DOD policy, security requirements, administrative controls, and technical measures to protect classified information processed by DOD computer systems. [DOD85] Meanwhile in the 1970's, the DOC, lead by its subordinate agency, the National Bureau of Standards (now the National Institute of Standards and Technology-NIST), began work to define problems and solutions for building, evaluating, and auditing secure computer systems.

The MITRE Corporation was tasked to develop a set of computer security evaluation criteria that could be used to determine the degree of trust an organization could place in a computer system to protect sensitive data. After much debate within the academic, industrial and government establishments, the draft of the initial TCSEC was produced.

The NCSC published the finalized TCSEC in 1983, and in 1985 the DOD published, with some revisions, this criteria into DOD Standard 5200.28-STD. From this original publication have come a series of guidelines and interpretations, each building upon or clarifying the original works of the TCSEC.

Subsequent to the incorporation of the TCSEC into a DOD standard, the NCSC began testing and evaluating products against the established technical standards and computer security criteria of the day. The NCSC maintains a list of evaluated products which it updates quarterly as part of the Information Systems Security Products and Services Catalogue. This Evaluated Products List (EPL) is a compilation of all computer products that have undergone formal security evaluations, and it shows the relative security merit of each product.

C. THE CRITERIA

The concept of the trusted computing base (TCB) is fundamental to the understanding of the TCSEC. (See "TRUSTED COMPUTING BASE" on page 29.) Once the TCB can be identified, evaluated, and rated, a level of assurance (rating class) can be given to the product and the system can be considered a trusted system. The Criteria contains three basic control objectives: security policy, accountability, and assurance.

1. Security Policy

The security policy of an organization is the starting point for any implementation of external and internal security mechanisms, and is a basic control objective of the TCSEC. The security policy must be defined in terms of the perceived threats, risks, and goals of an organization [DOD85]. The people or users of the system must be identified, and all the information that will be stored in the system must be located and distinguished from non-system information.

2. Accountability

Another control objective of the TCSEC is the accountability of subjects, which includes I&A and audit capabilities [DOD85]. Each access to a trusted system by a user must be mediated by a security control mechanism which correctly identifies individual subjects (by authenticating a password or other indelible unique feature) and controls what classes of information that subject can access to. A record of all security relevant actions (audit record) by the users must be kept so that any responsible party can be traced after a system violation has occurred.

3. Assurance

Assurance is the guaranteeing or providing of confidence that the security policy has been enforced correctly; that the reference validation mechanism does in fact do its job accurately by implementing the intent of the security policy. The security mechanisms which enforce the security policy must be capable of being "independently evaluated" so

that sufficient assurance can be given that the implementation does what the policy (or security model) promised. The Criteria specifies two components of assurance: life-cycle assurance and operational assurance.

Life cycle assurance is concerned with the way an vendor develops its products to be evaluated. A truly secure system must be built from the bottom-up, so that the during the design of the system, the product can be evaluated and tested at each major phase of the software development. It also suggests, that if a system is changed to the point that the integrity of the protection mechanisms are affected, then a reevaluation is required. This control objective leads to the RAMP (Rating Maintenance Phase) which requires all products to undergo re-evaluation when a new version of the product is released which significantly changes the system's security features. More will be said on this phase during the evaluation process to be discussed later.

Operational assurance focuses on the performance of the TCB and requires that the TCB be architecturally sound (i.e., process isolation, enforce least privilege, etc.,), periodically validated to be in correct operation, and run in a secure and correct way (i.e., facility management). Operational assurance also requires that if the computer system fails, that it can be brought back up in a secure manner.[DOD85]

In summary, according to the TCSEC, a computer system is "secure" if the following requirements are met:

- An adequate security policy (which includes access mediation) is defined and enforced
- All objects are properly labeled
- All subjects are correctly identified
- Accountability is maintained through audit capabilities
- The system can be evaluated and given an appropriate assurance level
- Continued protection of the TCB is maintained throughout its life-cycle

D. CRITERIA DIVISIONS AND CLASSES

The TCSEC is built to represent four different divisions for trusted assurance in computer products. The divisions from least restrictive (low assurance) to highly restrictive

(high assurance) are: D, C, B, and A. Overall there are seven different ratings (classes) that a product or system can earn; D,C1,C2,B1,B2,B3,A1.

1. Division D (Minimal Protection)

Division D contains one class (Class D), and is reserved for all computer systems or products that have failed to adequately meet the requirements of another higher evaluation class. Class D products or systems cannot be expected to protect any security policy or even human error.

2. Division C (Discretionary Protection)

Division C contains Class C1 and Class C2. These classes provide some confidence that the TCB is enforcing a discretionary security policy. The particular items of interest are discretionary protection, audit capabilities and verification and testing.

Tests must be conducted at Class C1 which verify that the security mechanisms (DAC, I&A) work in accordance with the system documentation. A level of assurance must be present that a user cannot by-pass or defeat the security mechanisms of the TCB. Additionally at Class C2, evidence must be demonstrated that I&A data and the audit data cannot be manipulated or destroyed by an unauthorized user. A search for obvious flaws must be conducted, so that any violations of resource isolation or unauthorized access to audit or authentication data is found. There must be "hands-on" involvement in the conduct of independent tests run by the evaluation team. [DOD85]

3. Division B (Mandatory Protection)

Division B contains three classes (B1,B2,B3) and introduces several new design requirements. Most significantly, this division introduces mandatory access controls, labeling of objects and subjects, covert channel analysis, and the requirement that the reference monitor concept be utilized in the Class B2 and B3.[DOD85]

It is at the Class B2 that serious security concerns are realized. At the lower classes of assurance, security can be thought of after-the-fact; an already designed system

can add-on mechanisms to be compliant with Class C1,C2, or B1 requirements. It is here at the Class B2, that security features must be designed into the system or product from the very beginning. It is for this reason that Class B2 is considered relatively resistant to penetration; the Class B3 is considered highly resistant to penetration.[DOD85]

Security testing must demonstrate that no subject can disrupt the TCB to the point that the TCB cannot respond to communications initiated by other users. [DOD85]

4. Division A (Verified Protection)

Division A contains one class (A1) and is characterized by the use of verifiable formal security methods or models. The Class A1 differs little from Class B3; they are said to be "functionally equivalent." The chief requirements for Class A1 are formal proofs must be developed which provide "a resulting high degree of assurance the TCB is correctly implemented."[DOD85]

All verification done at Class A1 must show that the formal top-level specification (FTLS) and the descriptive top-level specification (DTLS) are consistent with the TCB implementation. [DOD85]

For a summary of the assurance levels and their functional requirements, see Figure 11, below.

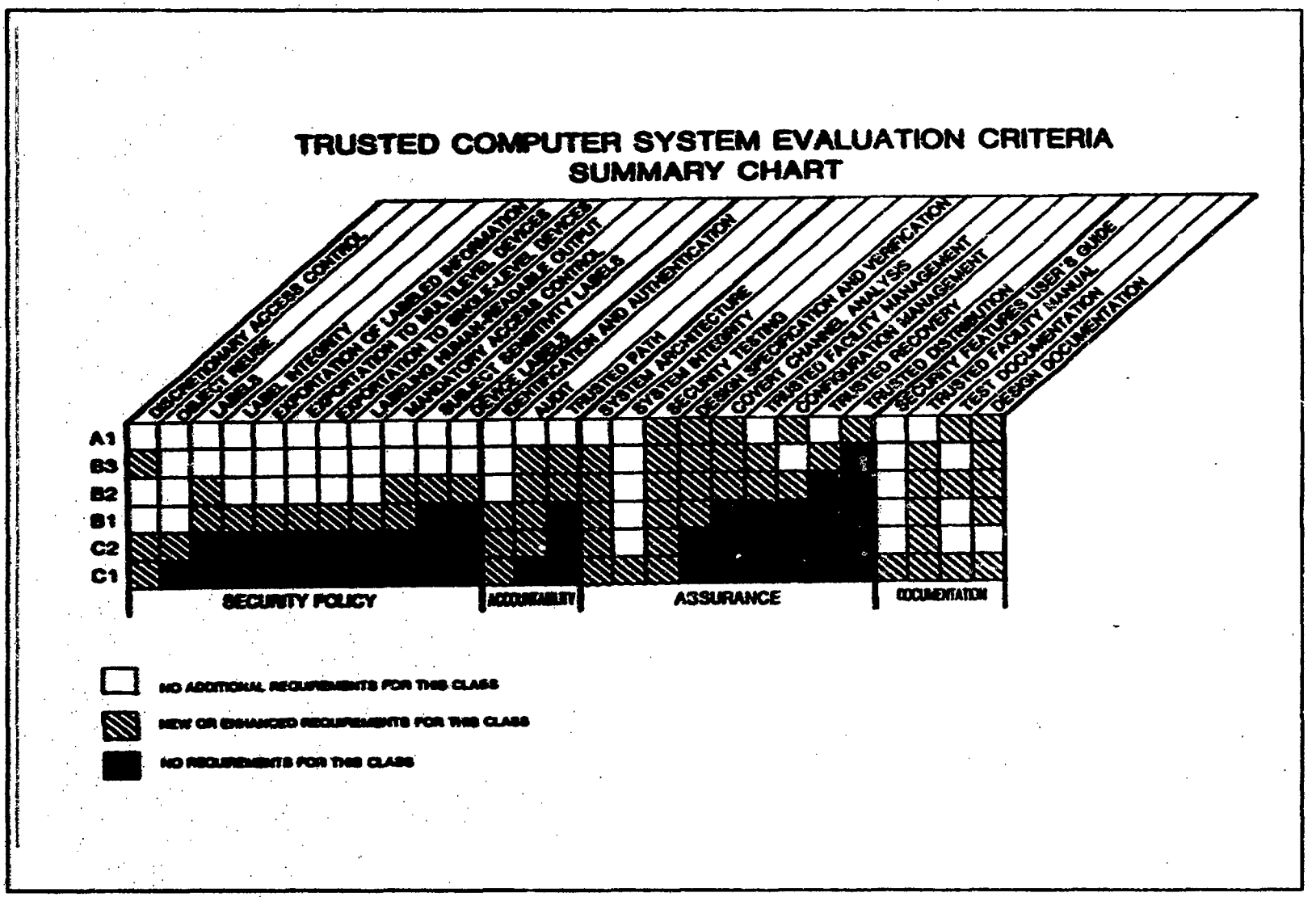


Figure 11: TCSEC Summary Chart from

E. THE DATABASE MANAGEMENT SYSTEM INTERPRETATION

The Trusted Database Management System Interpretation (TDI) of the TCSEC was issued as the third major interpretation to the TCSEC in April 1991 (the Trusted Network Interpretation (TNI) being the first in 1987 and the Computer Security Subsystem Interpretation (CSSI) in 1988 the second). The TDI extends the evaluation classes, described in the this chapter, to application products in general, and DBMS products in particular.

The TDI was produced to focus on the special problems posed by DBMSs and its purpose was to provide interpretations by which to build security features in DBMSs, to provide a metric for evaluating DBMSs, and to provide a basis for specifying security requirements in acquisition specifications. [NCSC91]

The TDI's central focus is the evaluation of a computer system comprised of parts. These parts, for example, could be the hardware, the operating system, or the application program (i.e., DBMS). The Interpretation is written in a general manner so that it can be used for evaluating all application programs, not just DBMS programs.

A interesting feature of the TDI is that it specifies that there is a difference between a security evaluation and a security certification or accreditation. A security evaluation is what is done by the NCSC in its Trusted Product Evaluation Program. An accreditation of a computer system or product is conducted by the using agency in the environment in which it operates. This so called "certification evaluation" is currently a big topic of discussion and research in the computer security community. The TDI may be used in both the evaluation portion and the accreditation portion of a system evaluation.

F. OTHER EVALUATION CRITERIA

The United States (U.S.) was the first country to have a trusted system evaluation standard which achieved widespread acceptance. [STRA93] However, since the TCSEC's inception in 1983, other countries have developed their own criteria for building and evaluating trusted computer products. Germany has the ZSIEC, France has the "Blue-

White-Red Book," Great Britain has the "Green Book" and Canada has the Canadian Trusted Computer Product Evaluation Criteria (CTCPEC). [TROY92]Since the birth of the European community as a political and economic entity, a more coordinated approach of defining computer security standards was needed. Four European countries (Germany, France, Great Britain, and the Netherlands) combined their resources to create the *Information Technology Security Evaluation Criteria* (ITSEC).[TROY92]

1. European ITSEC

Because the European community wanted to maintain commonality with the U.S., the members chose the TCSEC as a basis and elected to expand it, adding additional criteria and more detail. [TROY92] Version 1 of the ITSEC was published in June of 1990, and the second version released June 28, 1991. A number of evaluations have already been conducted against the ITSEC, including DBMS evaluations in the United Kingdom.

2. Canadian TCPEC

The Canadian Computer Security Establishment (CCSE) published the *Canadian Trusted Computer Product Evaluation Criteria* (CTCPEC) with influences from the Orange Book and the ITSEC.

3. Federal Criteria/Common Criteria

During the early 1990's, there began a move to consolidate a common federal criteria that would be more in-line with the proposed European ITSEC. The original goal of the Federal Criteria project was to create a U.S. national standard for computer and information system security that according to [CAMP94] would:

- protect previous investment in trust technology
- add value to current criteria
- develop a framework for defining new customer requirements
- promote international harmonization of criteria

This standard was intended to provide information on how to specify requirements for Information Technology product security, to include a fundamental

structure for stating those requirements and through a set of common building blocks to assist in the development process of the system.

The first draft of the Federal Criteria was released in December 1992, followed by national and international discussion between the U.S., Canada, and the European community. The Federal Criteria was then renamed the Common Criteria (CC) and subsequently a first rough draft has been completed. Comments from a very limited distribution list were due in the summer of 1994. This will lead to a revised draft which will be released to a wide audience in October 1994. [CAMP94]

It remains to be seen what the final form of the CC will be, and its effect on the TCSEC. According to NCSC, the TCSEC will continue to be the official standard against which evaluations will be made until the new criteria matures sufficiently [CAMP94].

IV. HP-UX BLS OPERATING SYSTEM

A. BACKGROUND INFORMATION

It is mandatory that we look at the operating system in our security analysis of both Trusted ORACLE and Informix On-Line/Secure. The reason is that the operating system is an extremely important subset of the trusted computing base. We have purposely chosen a common operating system, HP-UX BLS Version 8.0, so as to both limit the scope of the thesis (i.e., now only one operating system must be partially examined, instead of two), and to make our comparison of products have a common foundation (i.e., a common OS).

1. History

The UNIX operating system was developed by Ken Thompson of AT&T's Bell Labs in the late 1960's as a general purpose interactive timesharing system. After further refinements by the researchers at Bell Labs, UNIX became widely available in 1975. The University of California at Berkeley led the way in making many improvements to the system and began releasing their own improved versions called BSD (Berkeley Software Distribution). Meanwhile, AT&T continued to make improvements to their original system (System V) and thus released many new versions in the years to follow. These two versions, Berkeley's BSD and AT&T's System V, were in widespread use by the mid 1980's.

The standard Hewlett-Packard Unix (HP-UX) is based on and is compatible with UNIX System Laboratories (USL's) UNIX operating system [EDGE93]. USL's UNIX is similar to the Fourth Berkeley Software Distribution Unix software. Therefore, it has many of the characteristics of the Berkeley Unix operating system. However, HP-UX B-Level Security (BLS) does not support all the functionality of its predecessor, HP-UX. HP-UX BLS is a security enhanced version of HP-UX designed to meet the requirements of a Class B1 system [HEWL92a].

B. CONCEPT OF HP-UX BLS OPERATIONS

The HP-UX BLS operating system was enhanced to meet the criteria of a Class B1 assurance level, as described in the TCSEC. Therefore, with the exception of the new enhanced security features to qualify the system as Class B1 functional, the basic operating system is an exact version of the standard HP-UX system. (See "SECURITY ENHANCEMENTS" on page 49.)

1. Structure

The HP-UX BLS TCB consists of a modified HP-UX kernel, trusted commands and utilities, and trusted hardware and firmware [HEWL92a].

a. Kernel

The operating system's kernel is the most privileged part of the operating system which resides in the most privileged domain with direct access to the hardware. The kernel can be characterized as the implementation of the reference monitor concept; it is the most trusted piece of code which does the access checking between subjects and objects.

The HP-UX BLS kernel runs in the processor's protected mode and therefore runs in a separate domain of execution from that of the application software, which runs in user mode [HEWL92a].

b. Trusted Commands and Facilities

Trusted commands are all non-kernel programs that are responsible for performing special functions; they are trusted because they violate the security policy, but do so without resulting in unauthorized information flow. Facilities in HP-UX BLS are all libraries used to construct those programs responsible for security.

c. Trusted Hardware

Trusted hardware is that hardware which resides within the TCB and helps enforce the security policy of the system. Hardware specified by HP-UX BLS as part of the

TCB includes the processor and I/O internal buses, bus adaptor cards, disk drives, tape drives, and printers.

2. How UNIX Works

This is a brief overview of how UNIX works as taken from [TANE92].

A UNIX system, can be regarded as a kind of pyramid. At the bottom is the hardware, consisting of the CPU, memory, disks, terminals, and other devices. Running on the bare hardware is the UNIX operating system. Its primary function is to control the hardware and provide a system call interface to all application programs. These system calls allow user applications to create and manage processes, files, and other resources.[TANE92]

Application programs make system calls by putting arguments in registers and issuing trap instructions to switch from user mode to kernel mode to start up UNIX. A library is provided, with one procedure per system call. Each procedure first puts its arguments in the proper place, then executes the trap instruction. The trap instruction performs the required task and then returns to the user mode, where the application program is started again. [TANE92]

The operating system is a resource manager. It performs primitive functions to assist the application programs by controlling such things as the processors, memory space, and I/O devices.

C. SECURITY ENHANCEMENTS

As previously noted, HP-UX BLS is a security-enhanced version of the standard HP-UX operating system. A number of changes have been implemented to meet the Class B1 evaluation requirements. In the following sections, we will give a brief description of some of the security features that are implemented in the HP-UX BLS system.

1. Administrative Roles

One of the significant changes between standard HP-UX and HP-UX BLS is in the area of system administration [HEWL92a]. The system administration tasks have been split into a number of logical roles, thereby enforcing the concept of separation of privilege. The roles are split into functional areas, thus all of the roles can be given to one individual or they can be divided up between different individuals (depending on the needs and the

security policies of the using organization). Table 2, below, gives the different system administrative roles, along with the group names and responsibilities of each role.

TABLE 2: ADMINISTRATIVE AREAS FROM [HEWL92A]

Role	Group	Major Responsibilities
Authentication Administrator	auth	Sets up and maintains user accounts and parameters
Audit Administrator	audit	Runs and maintains the Audit subsystem
System Administrator	other	Configures new versions of the OS. Tunes system performance. Initializes the file system configuration
Subsystem Administrator	mem, backup, cron, terminal, lp, tape	Runs protected subsystems

2. Subsystem Features

Administrative roles are implemented through the mechanism of subsystem authorizations [HEWL92a]. A subsystem is defined by HP as a related collection of files, devices, and commands that serve a particular function. The subsystem authorizations allow the group which has access rights to the data in the subsystem to execute the subsystem. The only way a user can access subsystem information is by running programs in the subsystem [HEWL92a]. In Table 3, a list of the major subsystems are displayed along with the authorization required to execute the subsystems and the functions that each provide.

The chief rationale for providing subsystems is to enforce separation of mechanisms as noted in [SALT75]. When implemented correctly, one user can be prevented from having complete control over all resources within the system.

TABLE 3: PROTECTED SUBSYSTEMS FROM [HEWL92A]

Subsystem	Authorization	Function
Authentication	auth	Assigns authorization and clearances to users
Audit	audit	Maintains and analyzes output from the system's auditing functions
System Administrator	sysadmin	Configures new versions of the operating system and tunes the system
Memory	mem	Allows processes to read memory occupied by the operating system
Backup	backup	Maintains and backs up the file system
Cron	cron	Handles the scheduling of jobs on a delayed or periodic basis
Terminal	terminal	Controls terminal resources of the system
Line Printer	lp	Controls the printer resources of the system. Prints job requests made by users
Tape	tape	Controls the data import/export resources of the system

3. Login/Logout

The HP-UX BLS operating system requires the user requesting access to the system to enter his/her login name, password, and sensitivity level. This sensitivity label must be equal to, or lower than the user's clearance (the highest level the user is cleared for). The system then replies with the user's sensitivity level and the data (i.e, terminal ID) and time of last successful and unsuccessful login attempts.

The system administrator may select to permit user-defined passwords or may require the use of a random password generator. In addition, a password aging function can be selected which will prompt the user when it is time to change his/her password. If the password expires, then the user's account will be locked and the system administrator will be required to re-enable the account [HEWL92c].

4. Authorization and Privileges

An authorization is a right associated with a user; a privilege is a right associated with a process (or program) [HEWL92c]. HP-UX BLS implements kernel authorizations which give users the right to call upon the operating system kernel. However, if the program which calls the kernel does not possess the requisite privilege, then no action is taken by the OS kernel. The system administrator grants rights (i.e., base privileges) to users who are qualified by the security policy to receive them. In addition, the system administrator must also assign the right (i.e., granted privilege) to the program. Any user can attempt to run a program with an effective privilege set (the union of the base privileges and granted privileges) assigned to it, but if that user does not posses the kernel authorization in his own right then access will be denied.

The HP-UX BLS has one defined set of authorizations and four privilege sets. See Table 4 below.

TABLE 4: PRIVILEGE SETS FROM [HEWL92A]

Privilege Sets	Defined for	Description
Kernel authorization	processes	The set of rights for which a user is authorized. Trusted commands check a user's kernel authorizations before enabling a privilege.
Base privileges	Processes	Those privileges that are retained for all programs a user executes.
Effective privileges	Processes	Privilege against which all operational decisions are made.
Potential privileges	Files	The maximum set of privileges that the process running the program is allowed to use.
Granted privileges	Files	The set of privileges that are automatically added to the process's effective set when the program is executed.

The authorization set, although associated with a user, is stamped on all the user's processes. The above sets restrict users and programs in the use of system calls, and they create a mechanism which can be used to implement a policy of least privilege [HEWL92a].

5. Protecting Files

Protecting files in HP-UX BLS is similar to the protections found in standard UNIX, that is, the use of protection bits. (See "Protected Groups or Directories" on page 26.) In addition, because it is necessary to restrict file access to the granularity of a single user (which is not found in standard UNIX) to meet the Class B1 assurance level, access control lists (ACLs) are utilized in HP-UX BLS.

The ACLs are structured to provide three entries: user, group, and protection specification [HEWL92c]. The use of a special character "*", called a wildcard, enables general access to any user meeting the other requirements. Also, the protection specification can use r (read), w (write), x (execute), **all** (for r, w, e), or **null**, **none**, or "---" for no read, write, or execute. See Table 5 below for examples of how ACLs are used in HP-UX BLS.

TABLE 5: ACL ENTRIES IN HP-UX BLS FROM [HEWL92C]

ACL Entry	Explanation
<john.acct,r>	John, when in group **acct**, has read access
<*.acct, r-->	Any user, when in group **acct**, has read access but is denied write or execute
gary.*,null>	Gary, in any group, has no access permission
<*.progs,->	Any user in **progs** group is denied read access
<*.*,--->	Any user in any group is denied access

6. Mandatory Access Controls

Mandatory access controls (defined in Chapter II) are necessary for any system which seeks a Class B1 or higher assurance level. HP-UX BLS is designed to meet MAC requirements by labeling subjects and objects in the system with sensitivity labels and implementing a mandatory access control mechanism based on these sensitivity labels. Only users with special high-level privileges (e.g., the system administrator) can change the labels on some objects and this is done infrequently (this violates the tranquility property of the Bell-LaPadula model.);a regular user with default privileges cannot change labels.[HEWL92c]

The system administrator of the organization has the primary responsibility for setting up the sensitivity levels within the system according to the security policy. During installation, the system administrator sets the parameters within the MAC setup file (/etc/policy/mand/b1/mand_policy) to meet the requirements of the security policy. For example, the administrator will set the maximum sensitivity of information stored in the system at MAC_SYSHI, and the minimum sensitivity of information stored in the system at MAC_SYSLO. The maximum number of security levels (classifications) and maximum number of categories will also be set (default is 16 and 1024, respectively in HP-UX BLS).

The mandatory access controls allow subjects (processes) to read information at their sensitivity level and below, and to write to objects at their same sensitivity level. Files placed within directories cannot have a higher label than the directory since when a user creates a file it is written to its home directory. In addition, the user's home directory is set to at the lowest sensitivity level defined within the system (i.e., MAC_SYSLO).

The HP-UX BLS operating system maintains sensitivity labels for each subject and object, and this label consists of a combination of two components: a classification and a category set. A single classification is chosen from a hierarchical set of classifications defined by the administrator. For example, in the military context, this set might consist of TOP SECRET, SECRET, CONFIDENTIAL, and UNCLASSIFIED. The category set is

composed of zero or more nonhierarchical categories, which might include, (in military context) NATO, CRYPTO, and NUCLEAR. [HEWL92a]

7. Import/Export

HP-UX BLS controls all data imported into and from the system. The import medium is the magnetic tape or the floppy diskette; the export medium is the magnetic tape, floppy diskette, or printout. All import or export media are labeled or unlabeled. Labeled media are labeled with a sensitivity label that is recognized by the system (i.e., the system is set up to accept the label). The unlabeled media usually have external stick-on labels (or banner pages for a printout) that represent the sensitivity of the information on the medium.

All import/export devices are designated as either single-level or multilevel devices. Single-level devices are associated with a single sensitivity level and all data imported into or exported from the system is handled at that level. Multilevel devices can determine the sensitivity labels associated with objects imported to or exported from the system and then make the appropriate decisions to place the objects (i.e., files) in the correct directory or disk drive. (In the case of printout, it will place the correct label on each page of the printout by reading the file's label and then printing it on the respective page.) All device information is placed in a security database for retrieval by the access control mechanism.

8. Security Databases

HP-UX BLS requires the maintenance of several security databases to enforce the mandated security policy. (See Table 6 below). The protected subsystems access these databases when needed to obtain information for determining access control. The system administrator can change the parameters within the databases to suit the appropriate security policy.

All the security databases in Table 6 are self explanatory except for the System Defaults database. This database stores default values for the Protected Password, Terminal Control, and File Control databases. If the system administrator does not modify these three

TABLE 6: SECURITY DATABASES FROM [HEWL92A]

Databases	Contents
Protected Password	Contains each user's authentication profile
System Defaults	Default values for database fields
Terminal Control	Contains security information about each terminal
File Control	Contains protection attributes for each system file
Protected Subsystem	Contains security information about each protected subsystem
Device Assignment	Contains device-specific controls

databases, then the system will operate with defaults for such items as subsystem and kernel authorizations, password generation, and unsuccessful login attempts allowed per user or per terminal.[HEWL92a] By default, no ACLs are associated with objects, but the normal protection bits of UNIX still apply.

D. CONFIGURATION FOR DATABASE SUPPORT

Before a DBMS can be installed on an operating system a number of operating system specific issues must be addressed. In the following sections, we present some specific issues as they relate to HP-UX BLS and Oracle and HP-UX BLS and Informix.

1. Oracle Support

The Oracle utility, SQL*DBA, is used by the database administrator to set up the necessary operating system specifics prior to DBMS installation. The SQL*DBA calls the Bourne shell (command interpreter) utility program of HP-UX BLS as a default. If the other shell (i.e., C shell) is desired it can be specified. Both shells allow the DBA to call on the services of the operating system.

a. Memory Space Allocation

Input and output of both HP-UX BLS and Trusted ORACLE are done in units of storage called blocks. The size of Oracle blocks must be set by the DBA to enhance performance of the DBMS. It is recommended that a block size of 2K bytes be utilized upon initial installation. (The maximum Oracle block size is 8K).[ORAC92b]

b. Database and Log Files Size

The recommended database file size in Trusted ORACLE is 5 MB. A minimum of two log files is required per database and 100K bytes of storage is recommended for each file.[ORAC92b]

c. Filename Restrictions

Trusted ORACLE limits the length of some filenames to a maximum of 14 characters. [ORAC92b] This applies to HP-UX BLS file system when it does not have long filenames enabled within the system.

d. Terminal Characteristics

Some Trusted ORACLE utility programs (i.e, SQL*Plus) use special characters to call files which do not coincide with the characters found in HP-UX BLS. For example, the "@" character in Oracle calls an indirect command file; whereas this same character in HP-UX BLS is the line kill character default. This character should be redefined in Trusted ORACLE to avoid unexpected results.[ORAC92b]

2. Informix Support

Informix On-Line/Secure works with several secure operating systems, each of which has a slightly different implementation. Informix treats specific operating systems as if they are members of the following families: System V MLS, OSF MLS, CMW, and System V, version 4 ES. [INFO93b] HP-UX BLS belongs to the Compartmented Mode Workstation (CMW) family.

a. New UNIX Groups

The operating system administrator must set-up four new UNIX groups before installing OnLine/Secure. They are ix_data, ix_dbsso, ix_dbsa, and ix_users. The ix_data group is used by the operating system *root* account only, and allows the root account to maintain data within the database. The ix_dbsso and ix_dbsa groups are utilized by the database system security officer and the database system administrator respectively. All users of the database must be members of the ix_users group before they can access information within the database.[INFO93b]

b. Memory Space Allocation

The big decision for allocation of disk storage in Informix-OnLine/Secure is the decision to use a raw device or a cooked file for storage. A raw device is an area of disk that can be manipulated independently of the UNIX file system. A cooked file is a file whose disk space is managed by the UNIX file system. [INFO93b] Informix recommends the raw device option be used for storage because it enhances performance.

The page size for Informix-OnLine/Secure is machine dependent and fixed [INFO93d]. Therefore, since the HP-UX BLS page size is 2K bytes, so is the Informix page size. Even if the raw device option is used, a block size must still be specified for the Informix database to permit data import and export.

c. Shared Memory

Informix-OnLine/Secure uses shared memory and semaphores to allow different processes to share data and coordinate access to shared memory, respectively. When installing the DBMS, a number of parameters must be specified so that the operating system provides sufficient support for both shared memory and semaphores.

The shared memory parameter SHMMAX specifies the maximum size of a shared-memory segment and the parameter SHMSEG specifies the maximum number of segments a process can attach. Thus, given the needs of any specific user, the calculation SHMMAX * SHMSEG should be adequate for the required implementation.

V. TRUSTED ORACLE ARCHITECTURE

This chapter and the one following explain the configuration of the DBMSs and the HP-UX BLS operating system. Both systems can be configured in different ways, depending upon the choices made by the persons installing the system.

A. BACKGROUND

Trusted ORACLE 7 is a Class B1 security enhanced package based on the standard Oracle Relational Database Management System, Release 7.0. Therefore, the Trusted ORACLE 7 package includes all the features (functionality) of standard Oracle 7, along with multilevel security.[DATA94b]

1. History

Trusted ORACLE 7 is a distributed server database, first released in 1992. It includes all the functionality found in the standard Oracle 6 (its predecessor), plus a number of new features including a multi-threaded server, query optimizer, row-level locking, and role-based security. [DATA94b]

2. Platforms Supported

Standard ORACLE 7 can be installed on more than 88 different computing platforms [DATA94b]. However, Trusted ORACLE 7 is supported on only the DEC SEVMS and Hewlett-Packard's HP-UX BLS operating systems. Ultimately, the DBMS will be ported to a wide range of secure UNIX and proprietary platforms, including Compartmented Mode Workstations, as they become available from hardware and operating system vendors [EHRS91].

B. CONCEPT OF OPERATIONS

Trusted ORACLE 7 is a client/server architecture and can be configured in two basic modes: DBMS MAC mode and OS MAC mode. The following sections briefly discuss each DBMS mode.

1. DBMS MAC Mode

In the DBMS MAC mode database, only one database is created and this database contains multilevel data. This is in contrast to the OS MAC mode where multiple databases are created, one for each sensitivity level. (See "OS Mac Mode" on page 61.) In DBMS MAC mode, mandatory access control decisions are made in both the DBMS software layer and the operating system layer depending upon the object in question (e.g., file objects are controlled by the operating system, and database objects, such as tables and views are controlled by the DBMS). To accomplish the use of mandatory access controls, Trusted ORACLE 7 runs with special privileges that allow it to selectively bypass operating system security mechanisms [ORAC92a]. This makes DBMS MAC mode, according to Oracle, a trusted subject architecture [ORAC92a].

Figure 11 below, shows the DBMS MAC mode database and how each user, regardless of sensitivity level, connects to the database. The instance of the database (i.e., memory structures and processes running) is always labeled at the highest label in the database. Lower label users can connect to the database through this instance.

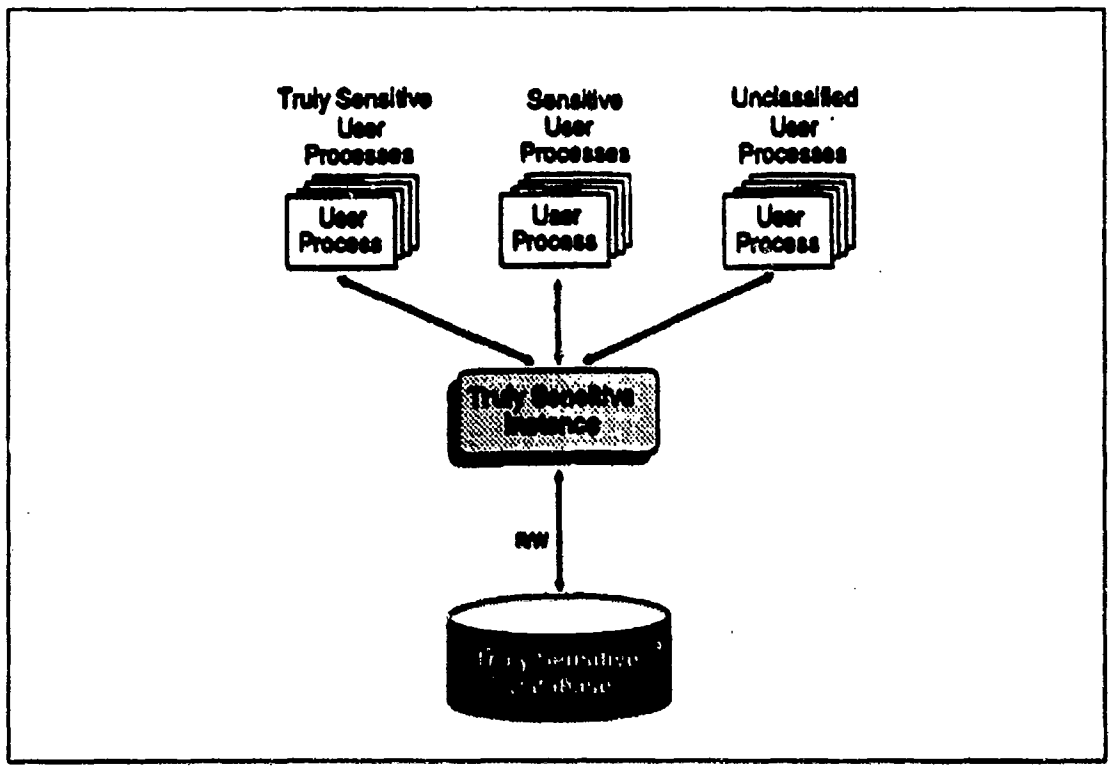


Figure 11: DBMS MAC Mode Database from [ORAC92a]

The DBMS MAC mode will be the mode that we investigate and analyze since it represents a multilevel database in which mandatory access control policy is enforced by the DBMS.

2. OS Mac Mode

In OS MAC mode, multiple, distinct, single-level databases are created, one for each sensitivity label. All mandatory access is mediated by the operating system on the operating system objects (i.e., files) in accordance with the overall security policy. Trusted ORACLE completely relies on the operating system to control access by Trusted ORACLE users to Trusted ORACLE objects. Multilevel tables can be created in OS MAC mode even though a single, physical table cannot contain rows of more than one label. A logical "multilevel" table can by created by identically named tables at each sensitivity label, each with identical attributes. Figure 12 below, demonstrates how a multilevel database system

is built using OS MAC mode. Note that three databases are necessary to represent a multilevel system with only three sensitivity labels. Because many separate databases are needed to represent many labels, one would use the OS MAC mode only when a small number of sensitivity levels are needed [ORAC92a].

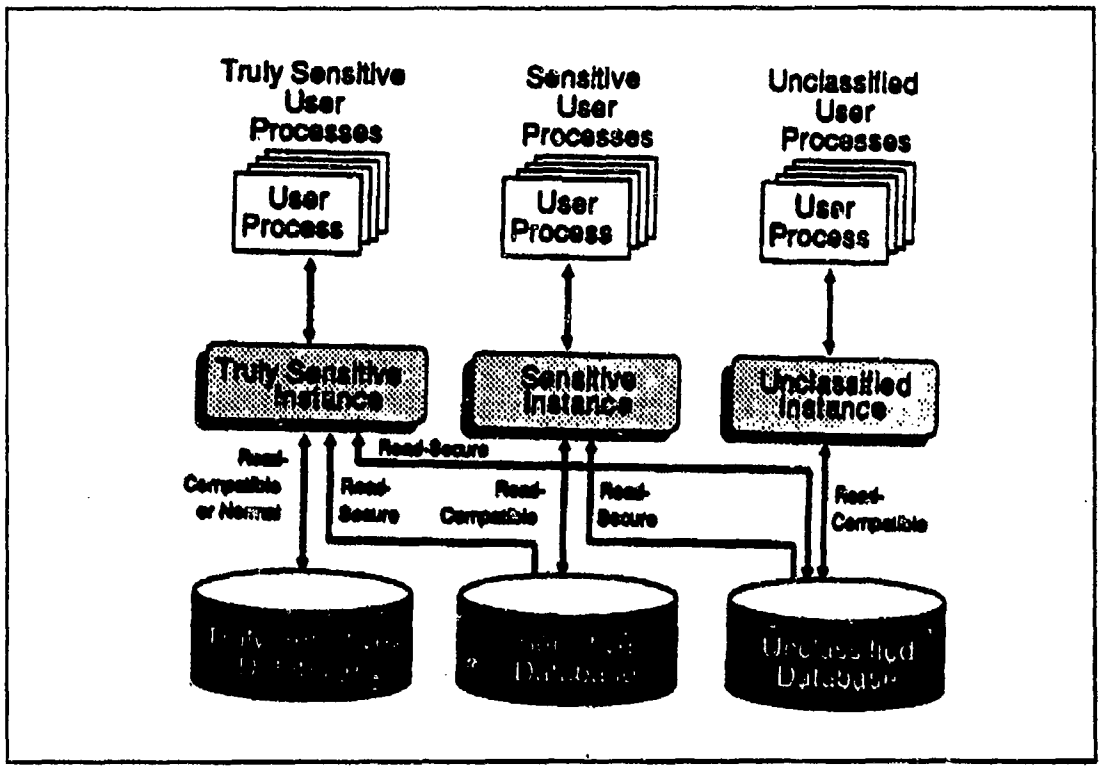


Figure 12: OS MAC Mode Database from [ORAC92a]

C. DATABASE STRUCTURES

The DBMS MAC mode database is similar to the standard Oracle 7 database. The logical structure is determined by one or more tablespaces and the databases' schema objects [ORAC92a]. The physical structure is determined by the operating system files which make up the database. All Trusted ORACLE databases are comprised of three types of files: one or more data files, two or more redo files, and one or more control files.

1. Physical Storage Structures

The following sections briefly describe the physical structures of an Oracle database.

a. Disk Organization

Trusted ORACLE can be set up to utilize raw disk devices. A raw disk device, or raw disk partition, is a hardware device that is supported by a character device driver. A character device driver accesses the raw device through special files that are in the */dev/rdsk* directory. These devices are not buffered by the HP-UX BLS kernel; data is transferred directly between the user's buffers and the device. Raw devices allow I/O directly between the disk where the data is stored and the System Global Area of the Trusted ORACLE server. The overhead of the HP-UX BLS read ahead and file system is avoided, thus performance is enhanced because data is stored together on the raw device.

b. Files

The data files are the files which contain the actual database data. Database schema objects (i.e., tables, clusters, indexes) are physically stored in the data files allocated to the database. A data file cannot change in size once created, therefore as a database grows in size, new data files are added to accommodate the database.

There are two or more redo files for every Oracle database. This set of redo files is known collectively as the "redo log." The redo log's primary purpose is to record all changes to the database. The information in the redo files is used only to recover the database from a system failure when the data has not been written to the data files.

One control file exists for every Oracle database. Its primary purpose is to record the physical structure of the database, such as the database name, the names and locations of the database's data files and redo files, and the time stamp of database creation.

c. *Memory Structures*

A process is a job or task that works within the memory of the computer. (A process is a subject as defined in Chapter II .) Figure 13 below, shows the important memory structures and processes in Oracle. The important structure to recognize is the "system global area" or SGA. The SGA is allocated anytime the database is started up, and the data within it is shared among the users currently connected to the database. The database buffers (i.e., database buffer cache) within the SGA, store the most recently used blocks of data; the redo log buffer stores the redo entries before they are written to the redo log files stored on disk. There is only one SGA per database instance.

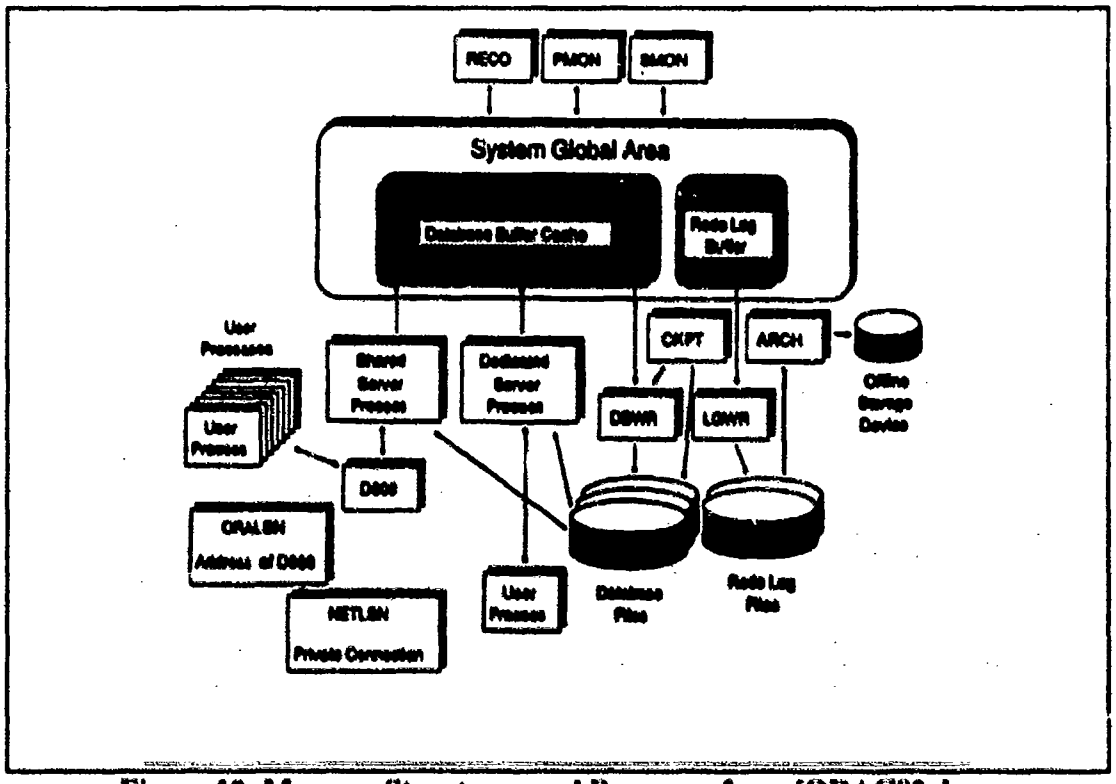


Figure 13: Memory Structures and Processes from [ORAC92c]

The Oracle RDBMS creates a number of daemon processes (background processes) for each database instance. These background processes perform certain

functions such as reads and writes to the database files (i.e., DBWR-Database Writer, and LGWR-Log Writer) and other needed checks and locks.

We have not ascertained from available documentation, what sensitivity level the daemons processes run at, or even if they have a sensitivity label at all. Other information (other than the *Trusted ORACLE User's Guide* and technical overviews) would have to be obtained to find the answers to this question.

d. Blocks, Extents, and Segments

The operating system file system has a specific number of bytes which make up an operating system block. In HP-UX, the block size is usually 2K bytes (2048 bytes). The Oracle database also recognizes, at its highest granularity level of storage, a data block (or page). Oracle allocates all its database space in blocks. This database block can be equal to the operating system block, or a multiple of it (e.g., a database block could be 2K or 4K bytes).

At the next level of storage is the "extent." An extent is a specific number of contiguous data blocks that are allocated for storing a specific type of information [ORAC92c]. For example, if more space is needed to store Oracle data files, then a data extent will be allocated for that data. If more space is needed for control information, then a control extent will be allocated.

The highest level of logical database storage is the segment. A segment is a set of extents which stores a specific type of data structure, such as a database table's data. The relationship between these three database spaces is shown in 14, below.

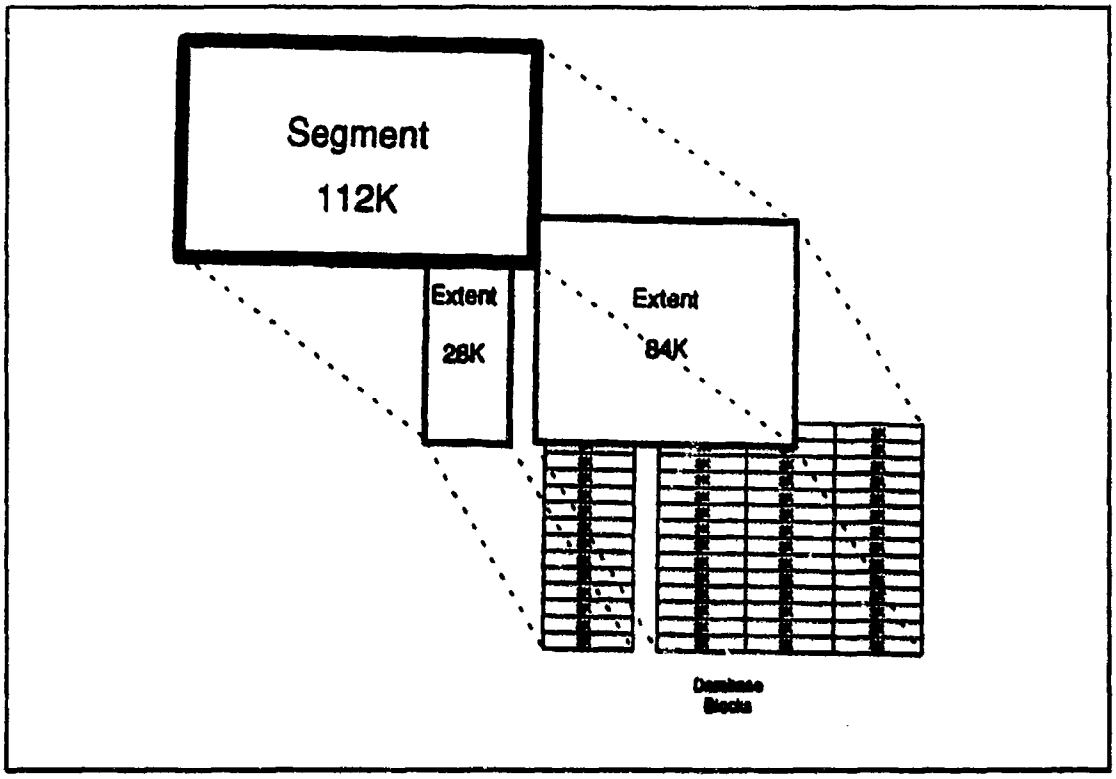


Figure 14: Data Blocks in Oracle from [ORAC92c]

2. Logical Storage Structures

The following sections briefly describe the logical structures of an Oracle database.

a. Tablespaces

The most important logical structures within the Oracle RDBMS are the tablespaces and the schema objects. Whenever an Oracle database is created, the system tablespace is likewise created. This system tablespace contains the data dictionary for the database and therefore must always reside in main memory. If more space is needed, then one or more data tablespaces can be added to the database. (See figure 15)

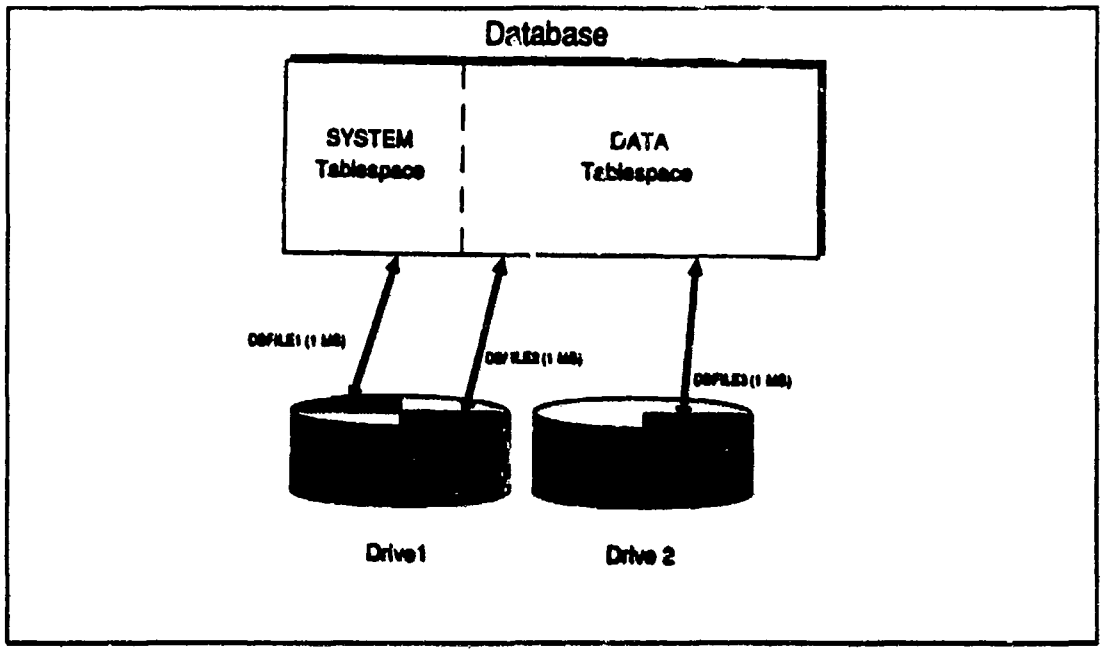


Figure 15: Logical Structures from [ORAC92c]

b. Schema Objects

Most schema objects such as tables, clusters, and indexes are stored within a tablespace. The data for a table is stored in one or more of the tablespace's data files. A cluster is an optional way of storing table data; it groups the tables that share the same data blocks together. This is because some tables share the same columns of identical data and are often used together. Clusters are used primarily to reduce I/O and to reduce the amount of storage space needed by storing redundant data only once.

A view is a tailor-made presentation of one or more tables and it is not stored within the tablespace or any other storage space. The only thing stored in a view is the view query or definition. When a view is invoked, it dynamically queries the appropriate tables stored in the database and then presents the data queried in a table-like format. (A view is often called a "virtual table.")

D. SECURITY ENFORCEMENT MECHANISMS

As previously stated, Trusted ORACLE 7 is a security-enhanced package of Standard Oracle 7 which has been designed to meet a Class B1 assurance level. In the following sections, we will briefly describe of few of the major security enhancements made to standard Oracle 7 to make it a Trusted ORACLE 7 - Class B1 complaint package.

1. Policy and Access Controls

The enforcement of discretionary access controls in Trusted ORACLE 7 is identical to those found in the standard Oracle 7 package [ORAC92a]. When user accounts are created in Oracle, no privileges are given to them by default. (This default rule of no access is consistent with the findings of Saltzer and Schroeder [SALT75].) The system administrator is responsible for giving the new users of the database what privileges they need based on their clearances, jobs, and the security policy being enforced.

Trusted ORACLE 7 has been designed to provide mandatory access controls, mediating access of labeled subjects to labeled objects. The MAC policy is an extended version of the Bell-LaPadula Model as discussed in Chapter II . The chief difference is that the Trusted ORACLE 7 policy does not allow a lower-level subject to write-up to a higher level object. Instead, a subject can only write to an object that has an equal label (i.e., the user's label must match that of the object.)

In Trusted ORACLE a MAC label consists of four components: sensitivity, integrity, information, and additional OS specific. However, most secure Class B1 operating systems do not support all these components (HP-UX BLS supports only the sensitivity component). The sensitivity component is made up of a single classification (i.e., sensitivity level) and zero or more categories. These classifications and categories are identical to those found in the underlying operating system.

Within the operating system, a label is a binary string. However, this binary string can be mapped to a numeric string or a character string to make reading the label easier. Trusted ORACLE supports a numeric format, and short and long character formats.

which are used to make labels human-readable. An example of a numeric might be 100:1, where 100 is the sensitivity level and 1 is the category. A short character format might be TS:A and a long format might be TOP SECRET:NATO. (More will be said about the Informix MAC features in the subsequent chapters.)

2. Privileges and Roles

A privilege in Oracle is a right to execute a particular type of SQL statement. Oracle divides privileges into distinct categories: system privileges, object privileges and MAC privileges (in Trusted ORACLE 7 DBMS MAC mode only.) System privileges allow users to perform particular system-wide functions, such as connecting to the database. Object privileges allow users to perform a specific action on a specific object, such as delete a row on the EMPLOYEE table. MAC privileges allow users to perform operations that circumvent MAC policy, such as reading higher level data.

Each MAC privilege corresponds to a similar privilege in the underlying operating system. A user with granted MAC privileges in Trusted ORACLE cannot execute the command successfully unless the corresponding privilege has been granted in the operating system[ORAC92a].

The three MAC privileges in Trusted ORACLE 7 DBMS MAC mode along with the needed HP-UX BLS privileges are shown in Table 7, below.

TABLE 7: MAC PRIVILEGES IN DBMS MAC MODE [ORAC92B]

MAC Privilege	HP-UX Privilege	Function
WRITEDOWN	downgrade or allowmacaccess	Allows users to perform write operations on data at a lower label.
WRITEUP	writeupclearance, writeupsyshi, or allowmacaccess	Allows users to perform write operations on data at a higher label.
READUP	allowmacaccess	Allows a user to perform read operations on data at a higher label.

The DBA grants MAC privileges to those individuals or roles which require their use. The MAC privileges do not override the operating system clearance defined on each user's account, but instead operates within that clearance. For example, a user with the READUP MAC privilege, can only read higher levels of information up to his/her OS clearance level.

3. Auditing

Trusted ORACLE allows the DBA to audit specific database objects, operations, users, and privileges [ORAC92a]. Two additional audits are recommended in DBMS MAC mode: covert channel auditing and MAC privileged operations auditing, for example when data is upgraded and downgraded. Auditing records in DBMS MAC mode can be sent to the database or the operating system audit trail [ORAC92a].

VI. INFORMIX-ONLINE/SECURE ARCHITECTURE

This chapter explains the configuration of the Informix DBMS. This chapter is our effort to explain the Informix DBMS structure so as to better prepare the reader for the subsequent comparative analysis of MAC policy enforcement.

A. BACKGROUND

Informix-OnLine/Secure is a multilevel secure relational database management system (RDBMS) for secure UNIX and compartmented mode workstation (CMW) platforms. OnLine/Secure comes in two different versions: B1 and C2. For the purposes of our comparative analysis, we only analyzed the B1 configuration.

1. History

Informix Software, Inc., is a subsidiary of Informix Corp., with corporate headquarters in Menlo Park, California. In September 1993, Informix's OnLine/Secure became the first database to meet Class B1 and Class C2 security levels, as specified by the NCSC (even though Final Evaluation Reports have yet to be made public as of this writing.)

2. Platforms Supported

Informix On-Line/Secure is available for Hewlett-Packard HP 9000 secure system, Sun Microsystems' Sun CMW secure system, SCO, Digital Equipment, Sun SPARC, and Zenith [DATA94b].

B. CONCEPT OF OPERATIONS

Informix-OnLine/Secure operates on a client/server model, where the client front end operates as a separate process from the server's backend process. These two processes

communicate via some form of interprocess communication (depending upon the platform).

Informix OnLine/Secure 5.0 implements multilevel rows within each database table. Each data record (e.g., row of a database table) is associated with the security level of the user who created or modified it most recently. Security clearances for users are defined by the operating system security officer when the users' accounts are created. Users determine their session level when they log into the operating system. This session level is dominated by the user's clearance. An individual user is only allowed to see or modify data which his/her session level dominates, unless special MAC privileges (called discrete privileges), are granted to the user. (See Chapter II for definition of dominates.)

In Figure 16 below, the basic architecture of the Informix-OnLine/Secure RDBMS is shown. Rows within their respective tables (i.e., Contracts Table) are segregated by their security classifications. For each table accounting information is maintained for all the security labels attached to each row within the table. All row data is placed in a logical storage space, called a bundlespace, as will be discussed shortly in the following sections.

The raw devices used by Informix-OnLine/Secure and the shared memory buffer cache appear to the UNIX operating system as single-level entities. However, the OnLine/Secure database treats them as multilevel storage. The RDBMS kernel is the entity which directly accesses the database raw devices and buffer cache. The database kernel is trusted to maintain the separation of objects at different security levels. All access to the database devices by the kernel uses the secure UNIX read, write, and seek functions. [INFO93a]

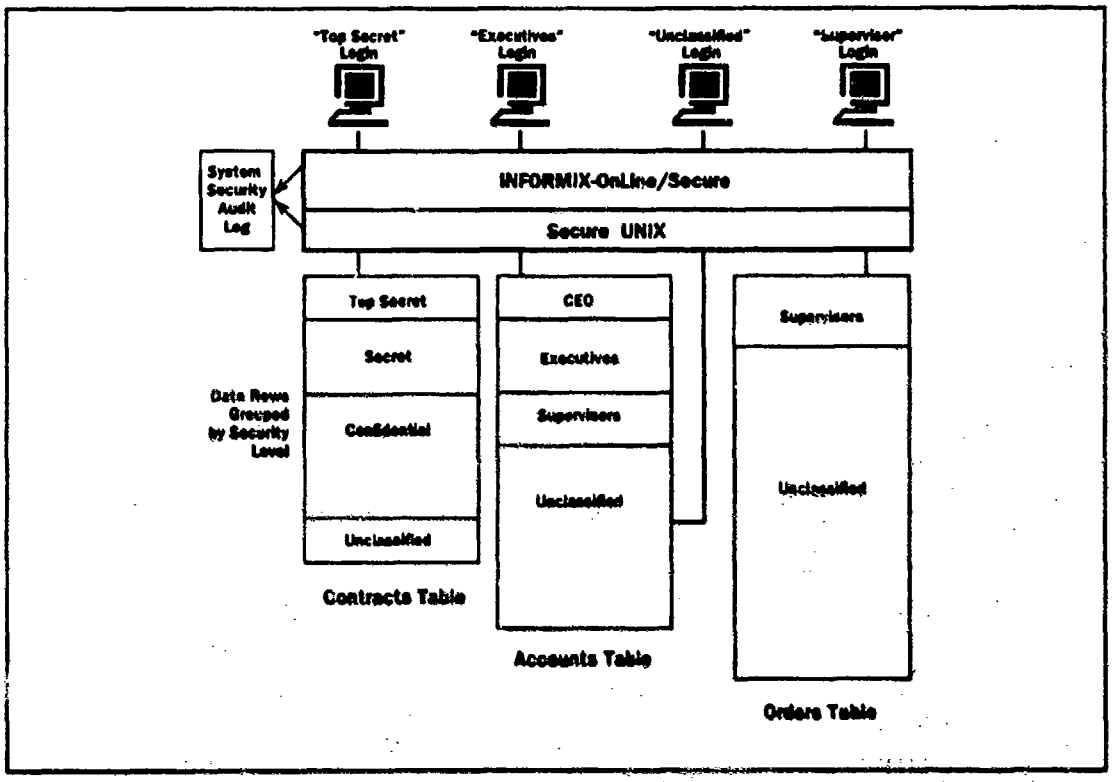


Figure 16: Basic Architecture from [INFO93a]

C. DATABASE STRUCTURES

Informix OnLine/Secure has additional structures and features relative to the standard Informix-OnLine RDBMS server. These new system capabilities are designed to meet the Class B1 assurance level of the TCSEC [INFO93c].

1. Physical Storage Structures

The following sections briefly describe the physical structures of the Informix-OnLine/Secure database.

a. Disk Organization

The Informix-OnLine/Secure database server is designed to perform its own disk management [INFO93d]. Raw devices are identified, (by using a UNIX utility), to be used in the storage of all database data and system catalogs. Raw devices are usually

physical devices, such as disk drives (or a partition of disk drive), and are carefully managed by the database server to enhance performance. (All raw devices for Informix are owned by the OS *root* account, with group ownership of **ix_data**, a special group necessary in Informix.) Initial storage space on these raw devices must be set aside prior to actually being used by the database server. To the greatest extent possible, OnLine/Secure bypasses the UNIX file system and works directly with the raw disk space.[INFO93d]

b. Shared Memory

Informix OnLine/Secure uses shared memory to hold database records (pages) in data buffers and to track information such as locks, active users, and open tables [INFO93a]. With shared memory, all programs (i.e., processes) that use the database access the same area of memory. (See Figure 17 below.)

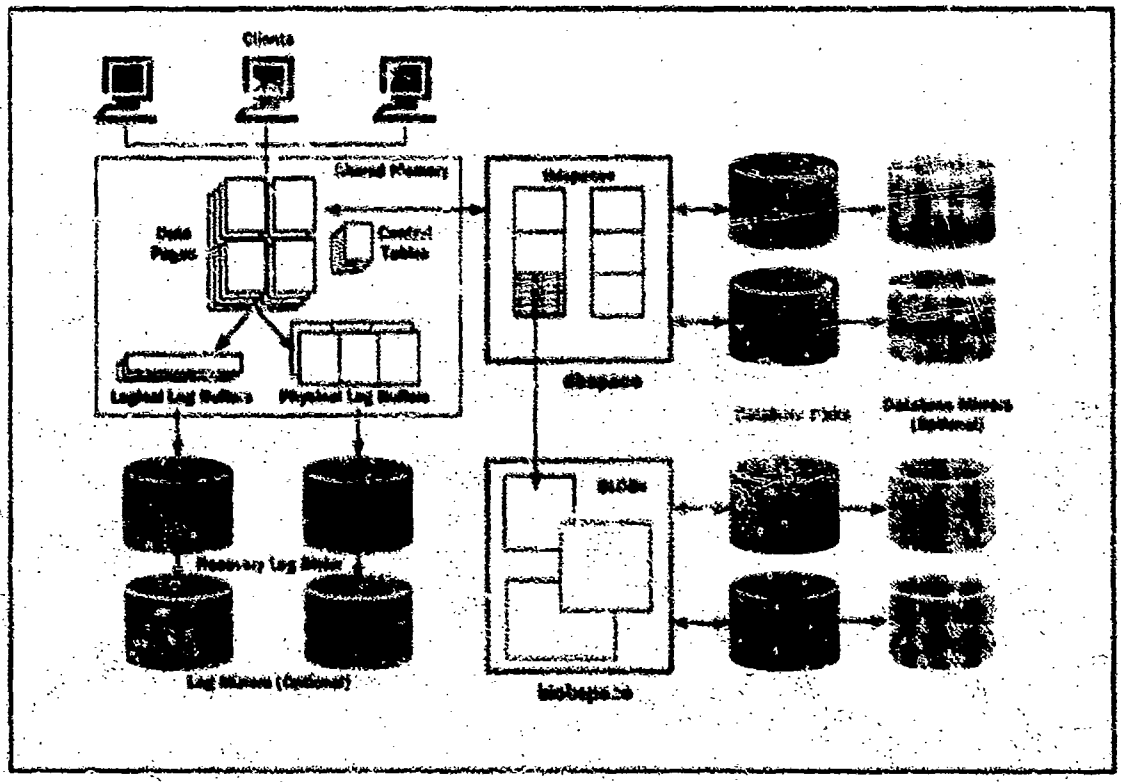


Figure 17: Shared Memory and Disk Structures from [INFO93a]

Shared memory is advantageous for several reasons, including the elimination of buffers for every process, (all database buffers are pooled), thus reducing disk I/O. Buffers are not reread, because only the most recent data page is in memory, and concurrency is enhanced because data is already in memory.

Logical logs record all the changes to the database since the last backup of data was made. The logical log buffers within the shared memory area are used to temporarily hold data before it is written to the logical log disks. The physical log buffers hold a copy of a database page on the disk before the page is changed. These "before-images" allow the system to reconstruct the state of the disk at the time of the last checkpoint (i.e., points in time when the database server knows all databases are consistent) before the system failure occurred.

Disk mirroring is the process of creating a mirror image of data in the database. This mirroring process requires the use of a primary database disk and a mirror disk. Database mirrors are optional in Informix-OnLine/Secure and are utilized for high availability.

c. Chunks, Pages, and Extents

The basic unit of storage in On-Line/Secure is the "chunk." A chunk is a unit of disk storage that has been dedicated to the Informix RDBMS server. Chunks can be either raw devices, parts of raw devices (i.e., partitions), or files under the UNIX operating system.

The page is the basic unit of disk I/O in the Informix database server. All space in every chunk is divided into pages and I/O is done in units of whole pages. The size of the page is the same in all chunks used for tables and is set when the DBMS is installed.

Informix OnLine/Secure allocates disk space on the raw devices in units called "extents." Each extent is a block of physically contiguous pages from the space designated to contain the database. When database users add rows to a table, or new tables,

and more space is required, an extent is allocated from the pool of available memory space by the DBMS for the new data.

The relationship of pages, extents, and chunks appear to be identical to those physical structures in Trusted Oracle (See Chapter V.) Chunks are made up of extents, which are added dynamically when more space is needed for data. All extents and chunks are in increments of database pages.

2. Logical Storage Structures

The following sections briefly describe the logical structures of an Informix-OnLine/Secure database.

a. Dbspace

When a database is created by the DBA or a standard user (i.e., SQL statement - CREATE DATABASE), it resides in a memory space called the dbspace. The dbspace is made of one or more chunks.

The root dbspace is always created first and must always exist because it holds the control information for other chunks that comprise the dbspace. The creator of a database can specify which dbspace to place the new database in; if no dbspace is specified, then a database is placed in the root dbspace.

b. Bundlespaces

The bundlespace was created by Informix specifically to accommodate multilevel data rows. All disk space (i.e., chunks) allocated to a specific table is accounted for in a bundlespace for that table. The bundlespace does not contain data and is used only for accounting purposes; it holds information about the individual tblespaces which hold all the data for the table. Bundlespaces do not contain security labels.

c. Tblspaces

There is a separate tblspace for each unique sensitivity level in the table. A tblspace holds all pages allocated to data (the rows) at one sensitivity level for a table, and

only pages at that level. If the table contained rows with three different sensitivity levels, there would be one bundlespace and three tblspaces for that table.[INFO93a] (See Figure 18 below.)

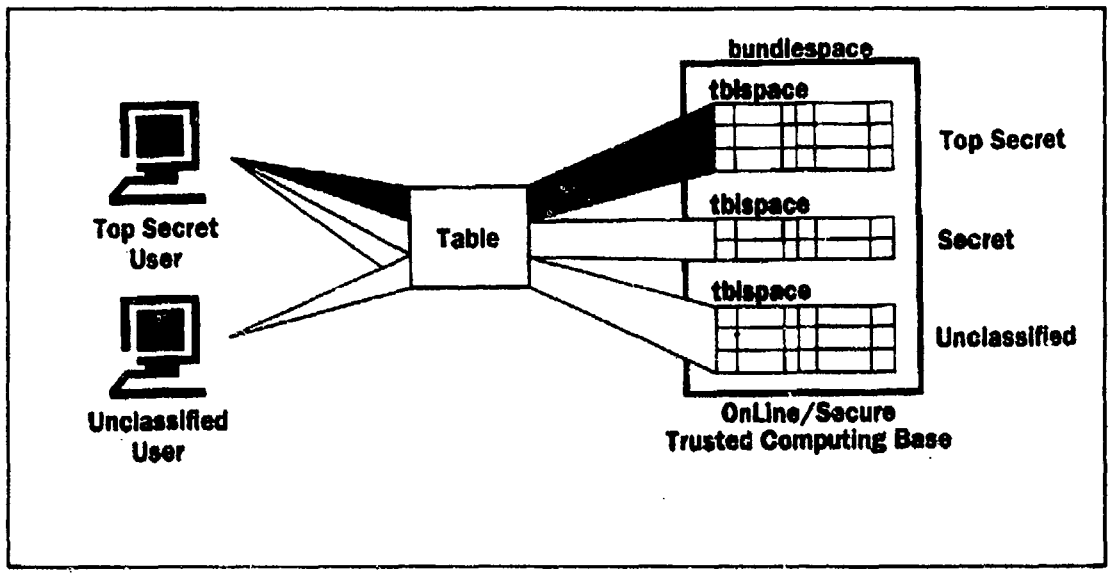


Figure 18: Tblspaces and bundlespaces [INFO93a]

In addition to data pages (which represent the data held in the rows of the table), the tblspaces also contain pages for indexes. Binary large object (blob) column pages are also found in the tblspace, even though the actual blob data is contained in a blobspace (which is similar to a tblspace, except it holds only special blob data types, see below.) [INFO93d]

Blobs are data storage objects that have no maximum size, except for the limitations of the computer, (usually 2^{31} bytes). Blob data types in Informix-OnLine/ Secure are TEXT and BYTE. The TEXT data type is used for storing ASCII data, and the BYTE data type is used for any type of binary data.[INFO93b]

d. Schema Objects

Schema objects in Informix-OnLine/Secure include databases, tables, rows, blobs, views, synonyms, indexes, contraints, and stored procedures. As previously

mentioned, tables and indexes (i.e., pages in their physical form) reside in tblspaces, which in turn reside in a dbspace. A table resides completely in one dbspace; if no dbspace is specified, the table resides in the dbspace where the database resides.

The schema object receives the same sensitivity label as the subject that created it. The label stays with the object throughout its life in the system, and only the database system security officer (DBSSO) can change it.[INFO93c]

D. SECURITY ENFORCEMENT MECHANISMS

Informix-OnLine/Secure has been designed to meet a B1 security assurance level. In the following sections, we will briefly describe a few of the major security features found in OnLine/Secure 5.0.

1. Policy and Access Controls

The discretionary access controls available in Informix-OnLine/Secure are the same as those found in Informix-OnLine. The DAC mechanisms give the owner of an object the ability to specify (using the SQL statements GRANT and REVOKE) which users can and cannot access data that he/she controls.

The mandatory access control policy of Informix-OnLine/Secure is an extended version of the Bell-LaPadula model. MAC breaks down into three simple rules: subjects can read only objects that they dominate, subjects can write only to objects at their security level, and object security levels do not change (except when the DBSSO changes them.) [INFO93c] (Chapters IX, X, and XI will explain the MAC policy further.)

Sensitivity labels in Informix-OnLine/Secure are composed of a hierarchical component and zero or more categories. These labels are the same set of sensitivity labels used in the underlying operating system. Advisory labels are sensitivity labels that are maintained by the non-mandatory TCB for convenience of the user. Therefore, if a database user request the sensitivity level of a database object (such as a row), the label returned to the user is an advisory label, not a sensitivity label.[INFO93c]

Sensitivity labels in Informix-OnLine/Secure are represented in four different formats: external, canonical (for System V MLS operating system only), internal, and tag. The external format is a human-readable label such as TOP SECRET:NATO; the internal format is a binary representation. (The canonical format is not used in the HP-UX BLS operating system.) The tag format is a 32 bit integer and is used extensively in the many operations performed on labels, such as label equality and label dominance. A tag is mapped to a human-readable label before exporting the sensitivity label to an output device such as a terminal or line printer, or it may be retrieved by a database user in the tag format.

2. Privileges

Privileges are used in Informix-OnLine/Secure to enforce discretionary access controls. There are three types of DAC privileges in Informix: database privileges, table privileges, and procedure privileges. All privileges are stored in the system catalog tables and any user with the "Connect" database privilege can query the system catalog tables to find out what privileges have been granted and to whom (assuming that this user's session sensitivity level dominates the information in the databases and system tables.) [INFO93c]

Database privileges from lowest to highest are C (Connect privilege), R (Resource privilege), and D (Database Administrator privilege). The Database Administrator privilege is not the same as the Database System Administrator (DBSA) privilege, which is given only to the database administrator. The Database Administrator privilege as mentioned here, allows users to execute the DROP DATABASE (i.e., remove a database from the system) and create DATABASE (i.e., establish a new database in the system) statements.

Eight privileges are applied to tables, which give non-owners the privileges of the owner. Table 8, below, describes each table privilege. A "-" indicates that a user does not possess the privilege; a capital letter, such as "S" allows the user to GRANT the privilege to another user; a small letter "s" does not.

TABLE 8: TABLE PRIVILEGES IN INFORMIX

Privilege	Symbol	Description
Select	s or S	Allows selection, including selecting temporary tables
Insert	i or I	Allows users to add new rows
Update	u or U	Allows users to alter existing rows
Delete	d or D	Allows users to delete rows
Index	i or I	Allows the user to create and alter indexes on the table
Alter	a or A	Allows users to add and drop columns; reset starting points for SERIAL columns
References	r or R	Allows users to specify referential constraints on a table
Column	*	Qualifies the Select, Update, and References privileges with the names of specific columns; allows specific access to those specific columns

The "*" in the Column privilege allows an owner of a table to grant another user the ability to read or update a specific column, while not reading or updating another column, within the same table.

The only procedure privilege is the Execute privilege, which allows the holder of this privilege to execute a previously defined procedure. When a procedure is created, only the owner can execute it; he/she must grant specific users the Execute privilege before they can use it.

Discrete privileges are used in Informix-OnLine/Secure to provide functions which do not adhere to the database security policy. These privileges allow users the ability to perform database operations that would otherwise be disallowed by the Informix-

OnLine/Secure MAC or DAC policies. [INFO93c] Discrete privileges allowed in Informix-OnLine/Secure are shown in Table 9.

TABLE 9: DISCRETE PRIVILEGES IN INFORMIX-ONLINE/SECURE

Privilege	Description
PRIV_CANSETLEVEL	Allows the user the ability to alter the session security level at which database operations occur
PRIV_CANSETIDENTITY	Allows the user the ability to alter the user name under which database operations are performed

The PRIV_CANSETLEVEL privilege enables a database user to successfully execute the SET SESSION LEVEL statement, thus effectively changing the session sensitivity level of the user. The PRIV_CANSETIDENTITY privilege enables the SET SESSION AUTHORIZATION statement so that a user can adopt the user name of any non-administrative user. (We will expound on these two privileges later in subsequent chapters.)

3. Auditing

The auditing records produced by Informix-OnLine/Secure events are stored in the operating system audit records only. All use of discrete privileges is audited in Informix-OnLine/Secure as well as all DBSSO actions, initiation of the database system administrator utilities, and each initiation of a new OnLine/Secure session. [INFO93b]

4. Secure Administration Front End

The DBSSO performs most of the security-related maintenance tasks using the secure administrator front end (SAFE). All auditing masks, MAC sensitivity labeling of objects, DAC privilege changes, and granting and revoking discrete privileges are done at the SAFE console. The SAFE provides an interface to the TCB and is part of the TCB. Only the DBSSO is allowed to perform operations at the SAFE.

VII. SECURITY ANALYSIS METHODOLOGY

This chapter discusses the rationale for choosing only certain TCSEC criteria to map to the DBMS implementations and gives a detailed decomposition of the chosen criteria.

The mapping methodology below, (See Figure 19) presents the overall methodology used to analyze the respective DBMS products. The TCSEC requirements are central to the analysis, and are located in the upper left hand corner. To the right of the TCSEC box is the Interpretations box; Interpretations would apply to any of the three primary interpretations that have been issued by the NCSC subsequent to the original release of the TCSEC military standard in 1985 (e.g., the TNI, CSSI[5], and the TDI). In our case, the TDI is the appropriate interpretation to be utilized. To the right of the Interpretations box is the Technical Manuals' box for each respective product. Actually, this box refers to any public documents that a prospective buyer could acquire prior to actually buying the software.

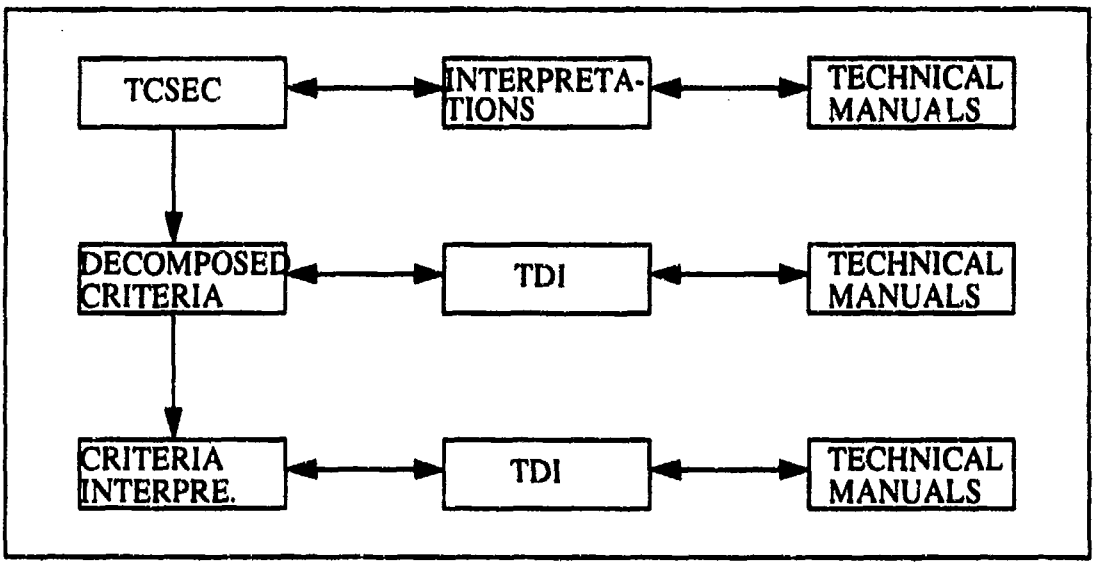


Figure 19: Mapping Methodology

5. CSSI - Computer Security Subsystem Interpretation of the Trusted Computer System Evaluation Criteria, NCSC-TG-009, Version-1, issued September 16, 1988.

The next level is basically a repeat of the upper level, except that the TCSEC requirements have been decomposed into more granular requirements. This allows more detailed analysis and a better understanding of what the overall criteria is trying to relay. Likewise, if substantive interpretations existed within the TDI, they to would be decomposed. However, as the present TDI exists, no substantive decomposition could be made.

The lowest level represents the decomposed criteria with individual "line item" interpretations thrown in. These "line item" interpretations are issued from time to time by the NCSC and published in the "Announce forum" of the Dockmaster bulletin board. We have only incorporated "line item" interpretations through September 1993, the time that the first DBMS (Informix-OnLine/Secure) completed substantive evaluation by NCSC. These interpretations are summaries and where found in *INFOSEC Handbook: An Information Systems Security Reference Guide* [ARCA93]. More recent interpretations would have to be accessed via the Dockmaster Announce forum.

Based on this simple methodology our analysis was conducted. Once both products are matched against the decomposed criteria and "line item" interpretations they will be compared. The comparison of DBMS products is found in Chapter X.

A. TCSEC CRITERIA CHOSEN AND WHY

Informix-OnLine/Secure 5.0 and Trusted Oracle 7.0 have both completed NCSC evaluation for Class B1 - Labeled Protection. The TCSEC Class B1 assurance level was discussed briefly in Chapter III, and will not be further expanded upon here. However, it should be noted, that the Class B1 level of assurance is characterized chiefly by the requirements for labels on some subjects and objects, a suitable MAC policy, and a mandatory access control mechanism implemented to enforce access by these labeled subjects to objects. Other new requirements do exist, such as design specification and verification, and security testing. However, we have characterized Class B1 assurance (labeled protection) as chiefly the implementation of a mandatory access control

mechanism against some labeled subjects and objects. The assurance provided at Class B1 is not significantly improved over that found in Class C2. Because of this reasoning and to focus this analysis, only the TCSEC Class B1 requirements for labels and mandatory access controls will be mapped to the respective implementations of Trusted Oracle and Informix On-Line/Secure.

B. CLASS B1 REQUIREMENTS DECOMPOSITION/SUMMARY

The following TCSEC Class B1 decomposed requirements have been extracted from the *INFOSEC Handbook* [ARCA93] and will be utilized to map the security features found within each product's DBA user's manual (and other relevant documents) to the overall requirement found within the Orange Book. The tables presented after each TCSEC decomposition is a summary of where the respective decomposed criteria were found. The following symbols are used in the tables:

- D - if the requirement was met "significantly" in the DBMS TCB component
- OS - if the requirement was met "significantly" in the HP-UX BLS TCB component
- B - if an "equal" combination of both the DBMS TCB component and the HP-UX BLS TCB component contributed to meeting the requirement
- U - users of the system are required to enforce this requirement
- NA - requirement does not apply
- NM - the requirement was not met in either the DBMS component or the OS component
- "*" - (asterisk) will be used if a TCB component exceeds the Class B1 requirement

A full discussion on the reasons for arriving at these symbols are found under their respective requirements in Chapter VIII for Trusted Oracle and Chapter IX for Informix-OnLine/Secure.

1. Key to Understanding Decomposed Statement Notation

The following table (See Table 10, below) constructed from [ARCA93], explains the notation used in the decomposed Class B1 criteria.

TABLE 10: DECOMPOSED CRITERIA NOTATION

Notation	Explanation
{ }	Text in braces replaces original TCSEC text, often done to replace a pronoun with its reference.
[]	Text in brackets is repeated from a previous criterion or is new text included for clarity.
...	Ellipses show were TCSEC text is omitted, typically done when a single TCSEC sentence divides into multiple criteria.
Italics	Italicized text denotes a TCSEC interpretation. Each of these criterion is followed by an interpretation number that generated it.

The TCSEC criteria interpretation summaries (in italics) are included adjacent to the specific criterion they affect. The NSA, over the years, has made a number of criteria interpretations (including discussion of alternate approaches, rationale, and presentation of the selected approach). The TCSEC criteria interpretations are independently numbered, with the assurance class, type interpretation, and the date the interpretation was published. For example, the LAB.i1 is referenced by C1-CI-03-89, which means that this interpretation starts at Class C1, is a criteria interpretation (CI), and was issued by NCSC in March 1989.

2. Labels

Labels are attributes associated with some subjects and objects in a Class B1 multilevel secure DBMS. These attributes represent the sensitivity or classification level of the subjects and the objects. The TCB is required to maintain these attributes for use by the access mediation mechanism.

Subjects are process-domain pairs. Daemon (background) subjects are maintained within the respective database server and operating system TCB components; each TCB component maintains its own set of daemon subjects. We will not analyze daemon subjects further due to a lack of appropriate documentation.

LAB.1 - Sensitivity labels associated with each subject... under its control (e.g., process...) shall be maintained by the TCB.

LAB.2 - [Sensitivity labels associated with each]... storage object [under its control] (e.g.,... file, segment, device) [shall be maintained by the TCB].

LAB.i1 - *Public objects shall be implicitly labeled with the minimum label in the system.* (C1-CI-03-89)

LAB.3 - [Sensitivity] levels shall be used as the basis for mandatory access control decisions.

LAB.4 - In order to import non-labeled data, the TCB shall request and receive from an authorized user the security level of the data...

LAB.5 - [In order to import non-labeled data, the TCB shall request and receive from an authorized user the security level of the data,] and all such actions shall be auditable by the TCB.

TABLE 11: LABELS SUMMARY

	Trusted Oracle Platform		Informix Platform	
Requirement	Oracle DBMS	HP-UX OS	Informix DBMS	HP-UX OS
LAB.1	B	B	NA	OS
LAB.2	B	B	B	B
LAB.i1	B	B	B	B
LAB.3	B	B	B	B
LAB.4	NA	U	U	NA
LAB.5	NM	OS	NM	OS

3. Label Integrity

Label integrity is concerned chiefly with maintaining the correct label on the respective subjects and objects, and ensuring that the TCB protects these labels from modification.

LI.1 - Sensitivity labels shall accurately represent security levels of the specific subjects or objects with which they are associated.

LI.2 - When exported by the TCB, sensitivity labels shall accurately and unambiguously represent the internal labels...

LI.3 - [When exported by the TCB, sensitivity labels]... shall be associated with the information being exported.

TABLE 12: LABEL INTEGRITY SUMMARY

	Trusted Oracle Platform		Informix Platform	
Requirement	Oracle DBMS	HP-UX OS	Informix DBMS	HP-UX OS
LI.1	B	B	B	B
LI.2	B	B	D	NA
LI.3	B	B	D	NA

4. Exportation of Labeled Information

From the TCB perspective, the exportation of labeled objects must maintain the integrity of the sensitivity label of the data with the I/O device which receives or transports the data out of the database.

EL.1 - The TCB shall designate each communication channel and I/O device as either single-level or multilevel.

EL.2 - Any change in {the single-level or multilevel} designation {of a communication channel} shall be done manually...

EL.3 - [Any change in {the single-level or multilevel} designation {of a communication channel}] shall be auditable by the TCB.

EL.4 - The TCB shall maintain... any change in the security level or levels associated with a communication channel or I/O device.

EL.5 - [The TCB shall]... be able to audit [any change in the security level or levels associated with a communication channel or I/O device.]

EL.i1 - *Level changes on single level communications channels and I/O devices shall be auditable. Level changes on multilevel communication channels and I/O devices are not required to be auditable.* C1-CI-01-88)

TABLE 13: EXPORTATION OF LABELED INFORMATION SUMMARY

	Trusted Oracle Platform		Informix Platform	
Requirement	Oracle DBMS	HP-UX OS	Informix DBMS	HP-UX OS
EL.1	NM	OS	NM	OS
EL.2	NM	OS	NM	OS
EL.3	NM	OS	NM	OS
EL.4	NM	OS	NM	OS
EL.5	NM	OS	NM	OS
EL.i1	NM	OS	NM	OS

5. Exportation to Multilevel Devices

Multilevel data must be exported to a multilevel device and the labels associated with the data must be correctly transported to the new medium on which the data will be stored. The new labeled data on the medium should correspond with the data as it was labeled in the TCB.

EM.1 - When the TCB exports an object to a multilevel I/O device, the sensitivity label associated with that object shall also be exported...

EM.2 - [When the TCB exports an object to a multilevel I/O device, the sensitivity label associated with that object]... shall reside on the same physical medium as the exported information...

EM.i1 - *Mulitilevel tape systems are not required to store an object's sensitivity label on the same tape as the object as long as this label can be associated with the object in a trusted manner.* (C1-CI-05-84)

EM.3 - [When the TCB exports an object to a multilevel I/O device, the sensitivity label associated with that object shall also be exported]... in the same form (i.e., machine-readable or human-readable form).

EM.4 - When the TCB exports... an object over a multilevel communication channel, the protocol used on that channel shall provide for the unambiguous pairing between the sensitivity labels and the associated information that is sent...

EM.5 - [When the TCB]... imports [an object over a multilevel communication channel, the protocol used on that channel shall provide for the unambiguous pairing between the sensitivity labels and the associated information that is]... received.

TABLE 14: EXPORTATION TO MULTILEVEL DEVICES SUMMARY

	Trusted Oracle Platform		Informix Platform	
Requirement	Oracle DBMS	HP-UX OS	Informix DBMS	HP-UX OS
EM.1	B	B	B	B
EM.2	NM	OS	D	NM
EM.i1	U	U	U	U
EM.3	NM	OS	B	B
EM.4	NM	OS	NM	OS
EM.5	NM	OS	NM	OS

6. Exportation to Single-Level Devices

The TCB must maintain single-level data exported to single-level devices by selecting the output device's sensitivity level based on the information being exported or imported. If data being exported or imported is SECRET, then the device chosen for the input/output should also be SECRET.

ES.1 - Single-level I/O devices and single -level communication channels are not required to maintain the sensitivity labels of the information they process.

ES.2 -... tne TCB shall include a mechanism by which the TCB and an authorized user reliably communicate to designate the single security level of information imported... via single level communication channels or I/O devices.

ES.3 -...[the TCB shall include a mechanism by which the TCB and an authorized user reliably communicate to designate the single security level of information]...exported [via single level communication channels or I/O devices.].

TABLE 15: EXPORTATION TO SINGLE-LEVEL DEVICES SUMMARY

	Trusted Oracle Platform		Informix Platform	
Requirement	Oracle DBMS	HP-UX OS	Informix DBMS	HP-UX OS
ES.1	NM	OS	NM	OS
ES.2	U	OS	U	OS
ES.3	NM	OS	NM	OS

7. Labeling Human-Readable Output

Human-readable output (i.e., paper reports, memos, etc.) must be properly marked with the correct label, as identified by the labeled object being exported.

HRO.1 - The ADP system administrator shall be able to specify the printable label names associated with exported sensitivity labels.

HRO.2 - The TCB shall mark the beginning and end of all human-readable, paged, hardcopy output (e.g., line printer output) with human-readable sensitivity labels that properly[6] represent the sensitivity of the output.

---

6. The hierarchical classification component in the human-readable sensitivity labels shall be equal to the greatest hierarchical classification of any of the information in the output that the labels refer to; the non-hierarchical category component shall include all of the non-hierarchical categories of the information in the output the labels refers to, but no other non-hierarchical categories.

HRO.3 - The TCB shall, by default, mark the top and bottom of each page of human-readable, paged, hardcopy output (e.g., line printer output) with human-readable sensitivity labels that properly[2] represent the overall sensitivity of the output or that properly represent the sensitivity of the information on the page.

HRO.4 - The TCB by default and in an appropriate manner, mark other forms of human-readable output (e.g., maps, graphics) with human-readable sensitivity labels that properly represent the sensitivity of the output.

HRO.5 - Any override of {human-readable sensitivity label} marking defaults shall be auditable by the TCB.

TABLE 16: LABELING HUMAN-READABLE OUTPUT SUMMARY

	Trusted Oracle Platform		Informix Platform	
Requirement	Oracle DBMS	HP-UX OS	Informix DBMS	HP-UX OS
HRO.1	NM	OS	NM	OS
HRO.2	NM	OS	NM	OS
HRO.3	NM	OS	NM	OS
HRO.4	NA	NA	NA	NA
HRO.5	NM	OS	NM	OS

8. Mandatory Access Control

The mandatory access control requirements address how labeled subjects access labeled objects, and if the access rules, (as stated by the security policy), are enforced by the MAC mechanisms.

MAC.1 - The TCB shall enforce a mandatory access control policy over all subjects... under its control (e.g., processes...).

MAC.2 - The TCB shall enforce a mandatory access control policy over all... storage objects {under its control} (e.g.,... files, segments, devices).

MAC.3 -... subjects and objects shall be assigned labels that are a combination of hierarchical classification levels and non-hierarchical categories...

MAC.4 -... {sensitivity} labels shall be used as the basis for mandatory access control decisions.

MAC.5 - The TCB shall be able to support two or more... security levels.

MAC.6 - The following requirements shall hold for all accesses between subjects and objects controlled by the TCB:

A subject can READ an object only if the hierarchial classification in the subject's security level is greater or equal to the hierarchial classification in the object's security level and the non-hierarchial categories in the subject's security level include all the non-hierarchical categories in the object's security level.

MAC.7 - The following requirements shall hold for all accesses between subjects and objects controlled by the TCB:

A subject can write an object only if the hierarchical classification in the subject's security level is less than or equal to the hierarchical classification in the objects security level and all the non-hierarchical categories in the subject's security level are included in the non-hierarchial categories in the object's sensitivity level.

MAC.8 - Identification and authentication data shall be used by the TCB to authenticate the user's identity...

MAC.9 - {Identification and authentication data shall be used by the TCB}... to ensure that the security level and authorization 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.

TABLE 17: MANDATORY ACCESS CONTROL SUMMARY

	Trusted Oracle Platform		Informix Platform	
Requirement	Oracle DBMS	HP-UX OS	Informix DBMS	HP-UX OS

TABLE 17: MANDATORY ACCESS CONTROL SUMMARY

	Trusted Oracle Platform		Informix Platform	
MAC.1	B	B	B	B
MAC.2	B	B	B	B
MAC.3	B	B	B	B
MAC.4	B	B	B	B
MAC.5	B	B	B	B
MAC.6	B	B	B	B
MAC.7	B	B	B	B
MAC.8	NM	OS	NM	OS
MAC.9	B	B	B	B

C. TDI INTERPRETATIONS

The Trusted Database Management System Interpretation of the Trusted Computer System Evaluation Criteria (TDI) was completed and issued in April 1991. One would expect that this publication might contain particular answers to questions related to trusted DBMSs. However, the TDI did not provide us with the revealing answers that we sought.

Section TC-5 of the TDI contains the "General Interpreted Requirements" for DBMS criteria. Often, the TDI added little other than a statement that the requirements of the TCSEC still applied.

For example, we have focused exclusively on Class B1 level Labels and Mandatory Access Control requirements for our evaluation and comparison of products. The general interpreted requirements for labels as stated in the TDI is:

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

This is generally the same type of interpretation that is present throughout the TDI. There is no "new" substantive interpretations found in the TDI that were of particular use to this researcher.

VIII. ORACLE ANALYSIS

This chapter analyzes the Trusted ORACLE 7 against the TCSEC requirements (as decomposed in Chapter VII) for labels and mandatory access controls. We start by looking at the DBMS TCB component (i.e., database server software and user's manuals), then proceed to the operating system TCB component. As discussed in Chapter VII, a requirement, as listed below, can be met in the DBMS TCB component, the operating system TCB component, both components, or it may not be meet or is not applicable. If users are required to meet this requirement, then a "U" will be placed in the respective columns associated with the requirement. After each requirement, we determine where the requirement was met, if in fact they were met. (See requirement summaries in Chapter VII.)

A number of the individual decomposed TCSEC requirements are substantially the same (some are exactly the same). Therefore, we will refer the reader to specified requirements in lieu of discussing the same requirement in two different places.

A. LABELS

The decomposed label requirements are discussed below.

1. LAB.1 Requirement

Subjects are the active processes in the system, be they user subjects or daemon (background) subjects. Daemon subjects are maintained within the respective database server and operating system TCB components; each component has its own set of daemon subjects which are created by the OS. Because we lack the appropriate documentation, we can not discuss daemon subjects specifically, and therefore will not analyze daemon subjects further.

The maintenance of a user subject's label begins with the creation of a username (i.e., account) for the Trusted ORACLE DBMS. All users must have a valid username before they can access the database. When an account is created for a new user on the Trusted ORACLE server, the account definition is stored as a row in a data dictionary table.

The row for the new username is labeled (under the ROWLABEL pseudo-column) at the same level as that of the database administrator (DBA in Trusted ORACLE) who created the new user. This requires the DBA who establishes the account to either connect to the database at the new user's projected label, or to "downgrade" or upgrade" his/her DBA label to match the desired label of the new user. Table 18, below is an example of an ALL_USERS table that maintains the user accounts in the Trusted ORACLE data dictionary.

TABLE 18: USERNAME DEFINITIONS IN TRUSTED ORACLE

ROWLABEL	USERNAMES	PROFILES
Secret	Ron	Level_1
Top Secret	George	Level_2
Unclassified	Dan	Level_3

The "PROFILES" attribute contains the set of specified resource limits that can be assigned to valid usernames and are used to prevent uncontrolled consumption of system resources. They are DBA defined in accordance with the security policy. These resource limits can be controlled at the session level, or the call level (i.e., when an SQL statement is executed), and include things such as CPU time, logical reads, concurrent sessions per user, idle time for a session, elapsed session time per session, etc.

The data dictionary table ALL_USERS, is stored in the SYSTEM tablespace, which is created automatically whenever the database is created. This SYSTEM tablespace is controlled by the Trusted ORACLE DBMS, which is a TCB component to the overall TCB of the entire system.

In the Trusted ORACLE configuration, it is mandatory that the underlying operating system (HP-UX BLS in this instance) maintain the user name and password used for identification and authentication. In the case of HP-UX BLS, new user names are set up by the Authentication Administrator (who is responsible for creating new users and

maintaining the Protected Password database within the operating system). Because authentication is performed by the OS, the user name and password of the operating system is the one used within Trusted ORACLE to authenticate that the user logging into Oracle is in fact a valid user. For example, if a user with an operating system account named "Ron" is to connect to the Trusted ORACLE database, there must be a corresponding database user "Ron" in the ALL_USERS table within the database data dictionary. When "RON" connects to the database (by typing "/"), the DBMS checks to see if there exists a valid user "Ron"; if so, then "Ron" can begin using the database.

Therefore, both components (the OS and the DBMS) are needed to maintain the user subject levels of the system and a "B" is given to each component in the summary tables in Chapter VII.

2. LAB.2 Requirement

As in the case of user accounts, all object definitions created in Trusted ORACLE are maintained as a row in a data dictionary. The objects found in Trusted ORACLE are database(s), tablespaces, rows, tables, views, indexes, clusters, sequences, synonyms, stored procedures and functions, packages, triggers, and rollback segments. The row for an object definition is labeled at the creator's label when the object is created. The data dictionary is located in the SYSTEM tablespace within the database. This tablespace is made up of segments which correspond to a set number of operating system blocks. Each tablespace is labeled when it is created. All objects placed within a tablespace must dominate the label of the tablespace. You cannot store a lower level object in a higher level tablespace [ORAC92a].

The storage objects seen by the HP-UX BLS operating system (i.e., files, directories, devices, IPC objects, symbolic links, named pipes, processes, printer queues, and ptys) are created and maintained in the OS. Files and directories are labeled individually, and are labeled in such a way that the access classes increase as you go down the tree. For example, the root directory is labeled at System Low and the directories and

files with the highest sensitivity labels will be found near the bottom of the file tree or in separate leaves off to the side of the tree. The UNIX file system maintains all the files and directories within the OS. All dynamic objects, such as printer queues, pipes and links, (and processes seen by HP-UX BLS) are also maintained in the OS TCB component.

Therefore, both components (the OS and the DBMS) are needed to maintain the object labels of the system and a "B" is given to each component in the summary tables in Chapter VII.

3. LAB.i1 Requirement

In this interpretation to the TCSEC, the requirement states that any objects which are to be accessed by all users of the system, shall be implicitly labeled at the lowest label defined for the system. We have not determined what implicitly labeling objects means, but instead will discuss how public objects can be explicitly labeled to meet this requirement.

If a user desires that an object be accessible to all database users, in effect making it a public object, then the creator must create the object at the lowest label defined within the database. In Trusted ORACLE this would be the equivalent of "DBLOW". The actual label for "DBLOW" is determined when the database is originally setup. In the military context, this would equate to the "UNCLASSIFIED" label with no categories. All unclassified objects are dominated (can be accessed) by all user levels in the database.

The same argument is used in the objects maintained by the operating system. If files and directories are to be public, both the files and their parent directories must be labeled at System Low in HP-UX BLS.

Therefore, both components (the OS and the DBMS) are needed to maintain that certain objects be public within the system and a "B" is given to each component in the summary tables in Chapter VII.

4. LAB.3 Requirement

This decomposed requirement is identical to MAC.4. (See "MAC.4 Requirement" on page 112.)

5. LAB.4 Requirement

The Trusted ORACLE DBMS utilizes an Import utility to import single level data, from an operating system file into the database. By default, the Import utility performs a single level import on a single level file. (A multilevel OS file can also be imported as single level.)The user importing the data has the responsibility for logging into the system at the level to match the data's label. Therefore, the user is responsible for meeting this requirement in the DBMS component.

The HP-UX BLS operating system handles the importation of Trusted ORACLE database files from outside the system. HP-UX specifically defines two types of import media, labeled and unlabeled. An unlabeled medium is one whose data does not include sensitivity labels. The unlabeled medium typically has some external label (such as a stick-on label for magnetic tape) which tells the user how to handle the data on the medium. This unlabeled medium must then be loaded on a single-level device (associated with a single sensitivity label) which corresponds to the label on the medium. When the data on this medium is loaded into the system, it is labeled at the same sensitivity as that of the single-level device.

Therefore, this requirement that the TCB shall request and receive the security level of the data, is accomplished directly by the authorized users of the system when they properly load tapes or floppy diskettes at the correctly labeled input device. Likewise, we label the HP-UX column of the summary table with a "U", and place an "NA" in the DBMS column since the Trusted ORACLE Import utility is only good if the files to be imported were created by the Oracle export utility.

6. LAB.5 Requirement

There is no mention of auditing the import of non-labeled data in the *Trusted ORACLE Administrator's Guide* [ORAC92a].

The HP-UX BLS operating system allows for the collection of audit data through the use of the Audit System Collection Mask. One of the audit capabilities of this

system mask is the auditing of subsystem events. Since import/export of data is a subsystem event (i.e., Tape), the HP-UX BLS system is capable of auditing all events associated with the import of non-labeled data.

Therefore, we place a "NM" in the Oracle column and an "OS" in the HP-UX OS column of the summary tables for this requirement.

B. LABEL INTEGRITY

1. LI.1 Requirement

The levels associated with subjects and the labels associated with objects are determined at the time of creation. It is not possible from the documentation as our disposal, to determine exactly how the TCB maintains and ensures that labels are attached to the objects.

It is up to the users who create the objects (the DBA in the case of creating subjects) to log into the system at the specified sensitivity level (i.e., to obtain a session level), so that when they create the objects, the TCB can correctly place this sensitivity level as an attribute value (of the row) where the object is defined in a system table. Therefore, we have noted in the summary table (See table 12 on page 87) that this requirement is met by a "B"; both components of the system are required to enforce this requirement.

2. LI.2 Requirement

Trusted ORACLE provides two label datatypes: MLSLABEL and RAW MLSLABEL. The MLSLABEL datatype is used to store a 4-byte internal tag that represents the binary format of an operating system label. Trusted ORACLE implicitly converts the operating system label placed in the ROWLABEL pseudo-column into a 4-byte tag. The RAW MLSLABEL datatype does not convert an operating system label into a 4-byte tag, but instead stores the actual OS label in binary format (up to 255 bytes). Either datatype can be specified.

The Export utility of Trusted ORACLE writes data from an Oracle database into operating system files in the Oracle binary format, either in the format of MLSLABEL or the RAW MLSLABEL [ORAC92a]. Within the Trusted ORACLE DBMS, the MLS keyword tells the Export utility whether or not to export labeling information along with the data a user is exporting. The default is Y (yes), which tells the Export utility to include the ALTER SESSION SET LABEL and the ROWLABEL pseudo-column. (The ALTER SESSION SET LABEL command ensures that when the export file is imported later, the imported objects are recreated at their original labels.) The ROWLABEL pseudo column values contain either of the two MLSLABEL datatypes.

This requirement is met by both the DBMS and OS TCB components, and is so reflected in the summary tables. (See table 12 on page 87)

3. LI.3 Requirement

As stated previously in LI.2, the sensitivity labels are associated with all objects created in the database (e.g., when a database table is created, the ROWLABEL pseudo-column, is automatically created as a special attribute) and tagged at the level of the user process creating the object. Therefore, when a multilevel export is conducted, the Export utility writes information to an operating system file, which includes labeling information for the data exported.

This requirement is met by both the DBMS and OS TCB components, and is so reflected in the summary tables. (See table 12 on page 87)

C. EXPORTATION OF LABELED INFORMATION

1. EL.1 Requirement

The functions to designate each communication channel and I/O device are found within the underlying operating system, HP-UX BLS. The system administrator defines the security characteristics of each import/export device by placing the required information in the Device Assignment database. Every device that is to be used must have

an entry in the Device Assignment database. Devices include terminals, line printers, and import/export devices such as tape drive systems and floppy drives. The operating system uses this database to restrict data objects that have sensitivity levels outside the range of the devices specified from either being sent or received through that device, thus preventing unauthorized disclosure. Every device is labeled specifically as either single or multilevel.[HEWL92a] Therefore this requirement is met by the OS TCB component only.

2. EL.2 Requirement

The system administrator, by invoking the *devasgif* command within the HP-UX BLS environment can change the parameters of the Device Assignment database. Only the system administrator with the proper kernel authorization and privileges can modify the Device Assignment database. The database contains information about all logical devices in the system, with each entry describing the devices characteristics (e.g., terminal, printer, removable media, such as a tape or floppy disk), whether its single-level or multilevel, or an import, export or both device. See Figure 20, below.

This requirement is met in that the system administrator with the proper authorizations and privileges is the only subject who can change the designation of a communication channel.Therefore this requirement is met by the OS TCB component only.

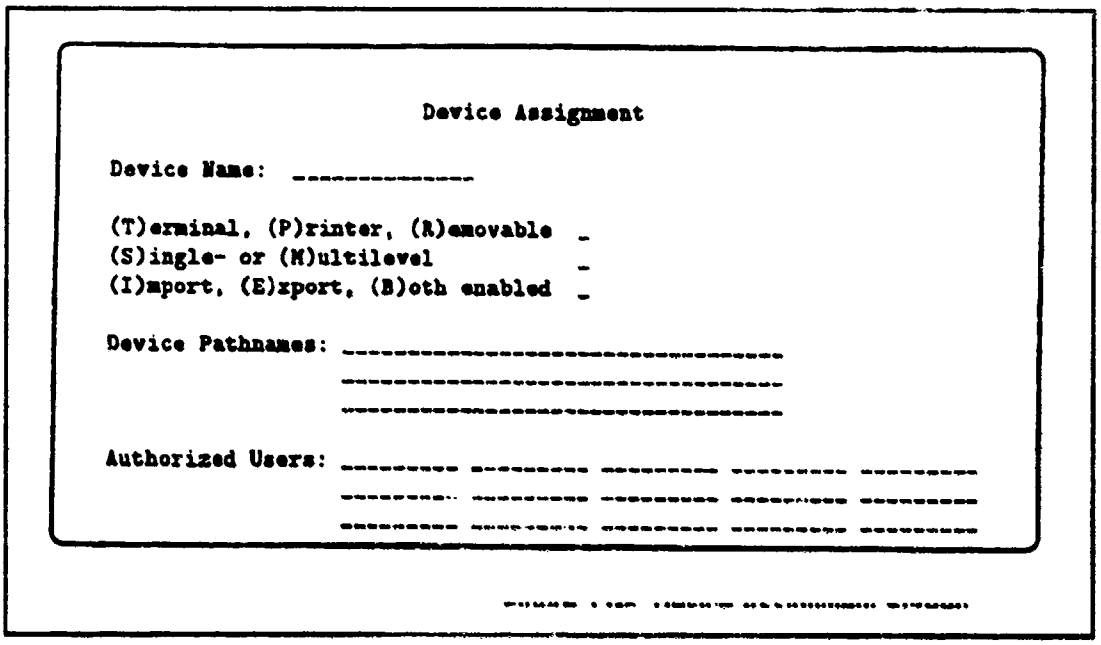


Figure 20: Device Assignment Screen [HEWL92a]

3. EL.3 Requirement

All Administrator/operator actions are auditable in they are found in the Audit System Collection Mask [HEWL92a]. The actions performed by the System Administrator, including the use of the *devasgif* command, are auditable. Therefore this requirement is met by the OS TCB component.

4. EL.4 Requirement

The TCB maintains the I/O device labels in the Device Assignment Database, a part of the HP-UX BLS operating system files. Any change to these security levels are found in the respective audit files or logs. Therefore this requirement is met by the OS TCB component.

5. EL.5 Requirement

This decomposed requirement is almost the exact requirement as stated in EL.3. The system administrator's action, if selected in the audit mask, will be audited. Thus, any

changes in the security levels associated with a I/O device are auditable. Therefore this requirement is met by the OS TCB component.

6. EL.i1 Requirement

Since all system administrator actions are auditable (if selected in the audit system collection mask), level changes in both single level and multilevel communications channels and I/O devices will be auditable. Though this interpretation states that multilevel communications channels and I/O devices are not required to be auditable, they will be in HP-UX BLS if the system administrator actions are selected in the Audit System Collection Mask. There is no granularity in the audit mechanism to preclude audit of certain system administrator actions. Therefore this requirement is met by the OS TCB component.

D. EXPORTATION TO MULTILEVEL DEVICES

1. EM.1 Requirement

This requirement is similar to LI.3. The Trusted ORACLE Export utility by default copies multilevel data to an operating system file. Then the *mltape* command within the HP-UX BLS operating system exports this multilevel data out of the system. The logic of this program deals with multilevel named objects, such as directories, and moves them to a tape device preserving all labels. If the user possesses the *multileveldir* kernel authorization, the hidden hierarchy of the directory (with all the hidden sub-directories) will be stored on the export medium, complete with the sensitivity levels of all hidden child directories (see Figure 21). [HEWL92a] (The MAC 0002, MAC 175, and MAC 1654 of the hidden directories in Figure 21, represent the sensitivity levels of the directories in a

numeric label format.) Therefore this requirement is met by both the DBMS and OS TCB components.

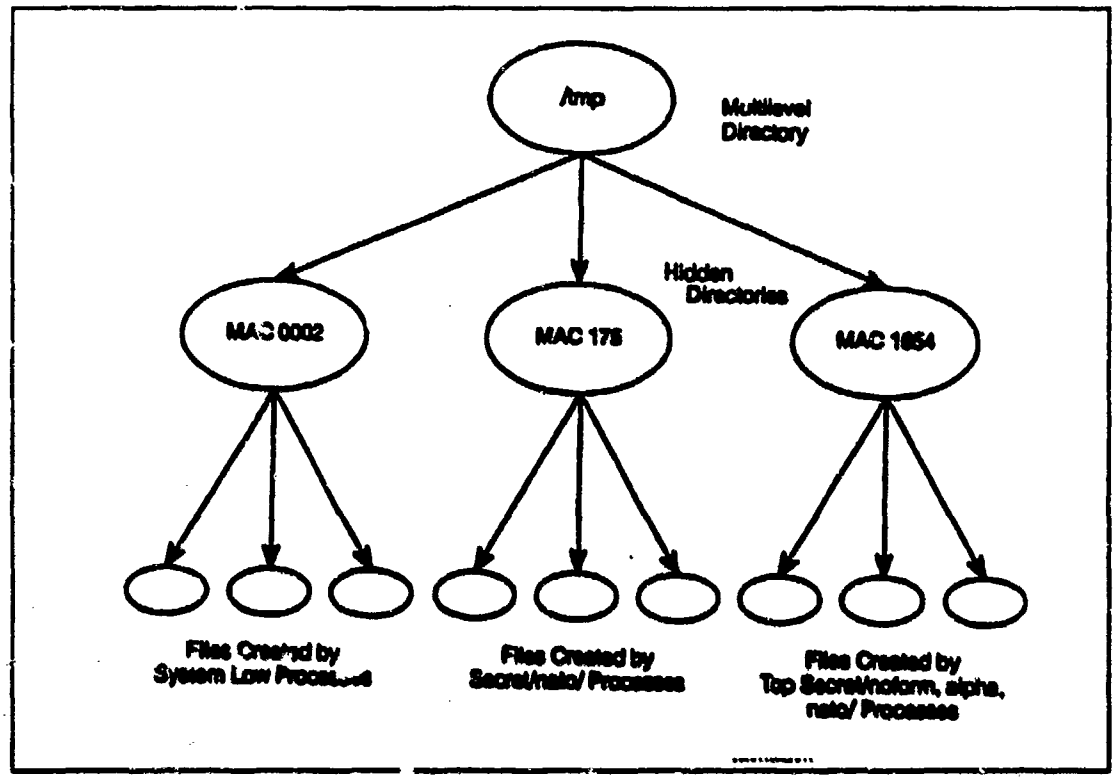


Figure 21: Hidden Directories in HP-UX BLS [HEWL92a]

2. EM.2 Requirement

The *mltape* command of HP-UX BLS correctly achieves the purpose of this requirement by copying the files into the specified device together with path name, status information, and security attributes. The security attributes of the file contain the security label. The operating system handles all copying to the output device. Therefore this requirement is met by the OS TCB component.

3. EM.11 Requirement

The Trusted ORACLE Export utility can be used to export data out of the database by specifying the single level export option on the command line (MLS=N). If a file created using the Export utility is later exported out of the system the *tar* and *cpio*

programs of HP-UX BLS can be called to export data to a single-level medium.The user is responsible to correctly label removable media if he/she transports unlabeled information out of the database. The label placed on the media must reflect the level of the classified information that the media contains. Therefore this requirement is considered a user requirement since the users of the system must properly label tapes and floppies with a label that reflects that of the data.

4. EM.3 Requirement

This requirement can be broken down again into two distinct requirements for analysis: one for machine-readable and one for human-readable. For machine readable form, the sensitivity label and the access control list (ACL) are placed on the tape in their extended form (ACSII), so that they may be read in by other similarly-configured systems.

For human-readable form, the sensitivity label produced on the banner page of the printout is at the sensitivity level of the user executing the print command. The user's sensitivity level is determined at the time they log-on the system.(See HRO.4 for more details.) Therefore, this requirement is met by the OS TCB component.

5. EM.4 Requirement

The MaxSix secure networking package is required for Trusted ORACLE [ORAC92b].

When the HP-UX BLS operating system is configured with the Trusted ORACLE server, additional networking software is required to setup the client/server and distributed database characteristics of Oracle. The SQL*Net Oracle software and the TCP/IP protocol adapter are required. Additionally the MaxSix MLS networking protocol is required by a Class B1 configuration.

The MaxSix protocol is a commercial trademark of SecureWare, Inc. and the company states that this protocol meets the requirements of the official U.S. Government standard, DNSIX 2.1 (DoD Intelligence Information Systems Network Security for Information Exchange) [ATKI94]. (Neither DNSIX nor MaxSix require any TCSEC class

level of assurance [ATKI94].) Therefore this requirement is meet by the OS TCB component when the MaxSix package is installed.

6. EM.5 Requirement

This requirement is met by the MaxSix MLS network protocol. See EM.4 requirement, above. Therefore this requirement is meet by the OS TCB component.

E. EXPORTATION TO SINGLE-LEVEL DEVICES

1. ES.1 Requirement

The HP-UX BLS operating system designates (through the operating system administrator) and maintains a device in the system as either single-level or multilevel. A single level device does not store labels with the files that it outputs. However, HP-UX BLS does specify a default sensitivity level for a single level device. This means that if a single level device is labeled, then only data at that level is exported through that device. Of course no labels will be associated with the exported files. Therefore this requirement is met by default in the operating system, and a NM is placed in the DBMS TCB component column.

2. ES.2 Requirement

The Trusted ORACLE Import utility is only good for files created with the Trusted ORACLE Export utility [ORAC92a]. By default, the Import utility performs a single level import on a single level export file. The user logs into the operating system at the level at which they want the information imported into the database. It is the responsibility of the user to know (based on the export file's label or other information) at which level to import the data. Therefore, for the DBMS component of the TCB, we have labeled this requirement as "U", for user responsibility.

The *tar* and *cpio* programs of HP-UX BLS have been modified to import single-level media.These programs perform the appropriate checks against the Device Assignment database to ensure that the device used for import is in fact specified as single-

level. Therefore, this requirement is met by the OS TCB and we labeled the summary tables column with an "OS".

3. ES.3 Requirement

The Trusted ORACLE Export utility writes information from the database to an operating system file at a single level (must select single level option, EXP / MLS=N). This single level export file contains no labeling information. For example, in Trusted ORACLE, you can export a multilevel table as a single level export; the resulting export file contains no labeling information and is consequently single labeled at the user's session level [ORAC92a]. The user's session level must dominate all data if it is to be exported to the export file. If a user logs in at SECRET, then all information at SECRET and below is exported to a single-level file (with no labels). The export file is then exported out of the system by the appropriate OS programs.

The *tar* and *cpio* programs of HP-UX BLS have been modified to export single-level files. These programs perform the appropriate checks against the Device Assignment database to ensure that the device used for export is in fact specified as single-level. Therefore, this requirement is met by the OS TCB only.

F. LABELING HUMAN-READABLE OUTPUT

1. HRO.1 Requirement

The operating system administrator only indirectly specifies the printable label names associated with exported security labels. Labels names are set-up when the sensitivity classifications and categories are defined for the system. HP-UX BLS prints the sensitivity label of the process executing the print command on the banner page of the printout. This is the sensitivity label that was typed when the user logged into the system (i.e., session level). Therefore, this requirement is met by the OS TCB when the system administrator set up the label names.

2. HRO.2 Requirement

The HP-UX BLS prints the sensitivity level of the process executing the print command on the banner page of the printout. It usually appears in the same location on the banner page as the print-job detail (i.e., filename, process number, date, etc.). As stated in HRO.1, this banner page label is the same as the user's session level.

There is no mention of marking the end of all human-readable paged, hardcopy output with a trailer page. However, the last page printed (a body page) will have a human-readable sensitivity label printed on the top and bottom of the page. Therefore, this requirement is met by the OS TCB.

3. HRO.3 Requirement

When the information is printed on the printer by the HP-UX BLS operating system, the banner page includes the sensitivity level of the process and each internal page includes the sensitivity level of the file that appears on that page. Internal pages (body pages), are labeled with the highest sensitivity level of the information that is printed on the page.

Top and bottom labeling is characteristic only of the body pages of the printout. The banner page only prints the classification one time, usually near the print-job detail. There is no mistaking a banner page with a body page in HP-UX BLS, because the banner page is uniquely designed and standard. Therefore, this requirement is met by the OS TCB.

4. HRO.4 Requirement

This requirement does not appear to be applicable in HP-UX BLS operating system. Labeling is supported on the line printer only. Labels are not supported on laser printers or plotters [HEWL92c]. In addition, labeling is not supported when the output is assigned a "landscape" orientation (i.e., the output is printed horizontally)[HEWL92c]. However, since HP-UX BLS does not support these printing options (it only prints line printer text) this requirement is not applicable to the OS TCB or the Trusted ORACLE TCB components.

5. HRO.5 Requirement

HP-UX BLS provides two secondary line printer subsystem options which can override the labeling and filtering capabilities of normal printing. The *label* authorization allows the use of the *-l* option to the *lp* command to suppress the labels applied to the internal pages of a printout. The *filter* authorization allows the *-f* option to the *lp* command, which removes the filtering done on printed output. Both of these options are audited by the system [HEWL92a]. Therefore, this requirement is met by the OS TCB.

As a side note, both the special options to the *lp* command suppress the internal printout labeling features of HP-UX BLS. The assumption is that the banner page is still printed.

G. MANDATORY ACCESS CONTROL

1. MAC.1 Requirement

The Trusted ORACLE DBMS TCB component enforces a mandatory access control policy over all users (and their surrogate processes) through the levels attached to each user's process and the label attributes attached to each object. Essentially, the MAC policy used in Trusted ORACLE is an extension of the Bell-LaPadula security model's mandatory access controls. (See Chapter II for the Bell-LaPadula Model.)This MAC policy, based on reading objects and writing objects by subjects, is discussed in depth in the MAC.6 and MAC.7 requirements below.

The HP-UX BLS TCB component maintains sensitivity levels on all subjects (active entities in the system, such as processes). When a user logs into the system, a login user ID (LUID) is attached to the user's login process. The LUID points to the Protected Password database which contains the clearance level for that particular subject. (The clearance level is the highest sensitivity label allowed for that subject's process.) All processes spawned from this LUID process contain this same LUID with its associated clearance level. From these subject processes' levels a mandatory access control policy, based on the Bell-LaPadula security model, is enforced. (See the MAC.6 and MAC.7

requirements for details below.) Therefore, this requirement is met in both the DBMS TCB and the OS TCB.

2. MAC.2 Requirement

The Trusted ORACLE DBMS TCB component enforces a mandatory access control policy over all objects controlled by the DBMS through the labels attached to each object. This MAC policy, based on reading objects and writing objects by subjects, is discussed in depth in the MAC.6 and MAC.7 requirements below.

The HP-UX BLS TCB component maintains sensitivity labels on all objects controlled by the operating system. These objects include regular files, inter-process communication (IPC) objects, directories, special files, pipes, processes, and symbolic links. When objects are created, the HP-UX BLS system attaches security attributes to the objects. For example, when a file is created, the full pathname of the file, the file owner and group, the file mode and type, the sensitivity level, the potential and granted privilege sets, and the access control lists are stored in the File Control database. Everytime a process attempts to access the file, it must search the File Control database, check, and pass each parameter before access is granted. The MAC policy for reading and writing these objects is discussed in depth in the MAC.6 and MAC.7 requirements below. Therefore, this requirement is met in both the DBMS TCB and the OS TCB.

3. MAC.3 Requirement

Within the Trusted ORACLE TCB component, subjects and objects are assigned sensitivity levels at the time they are created. Trusted ORACLE sensitivity labels for both subjects and objects consist of four components. See Figure 22 below.

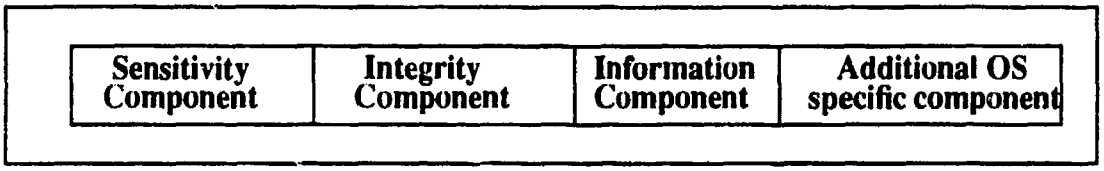


Figure 22: Trusted ORACLE sensitivity labels

The sensitivity component consist of a single classification, which has been previously defined by the system administrator, and zero or more categories. This range of sensitivity classifications and the categories allowed in this block match those of the underlying operating system. (The TCB as a whole can only have one set of classifications; all TCB components must recognize and accept these classifications.) The integrity component of the label reflects the object's trustworthiness or accuracy. The information component is used when a compartmented mode workstation (CMW) environment is utilized, or it may, in some systems be used as advisory labels. The last component, is used for any operating system specific component that can be utilized by the operating system.

The HP-UX BLS operating system does not support the integrity component or the information component of the Trusted ORACLE sensitivity label. They are therefore not utilized when the TCB consists of the HP-UX BLS platform and the Trusted ORACLE DBMS. The HP-UX BLS system specifically supports the U.S. DOD method of classifying information according to hierarchial classification levels and non-hierarchial categories. Therefore, this requirement is met in both the DBMS TCB and the OS TCB.

4. MAC.4 Requirement

Mandatory access controls are enforced through the labels attached to each subject and object. Trusted ORACLE provides MAC mediation access based on the identity (e.g., LUID) and label (clearance) of the subject and the sensitivity or label (classification) of the object. We can only visualize how this works since we are not privy to the internal data structures of the Trusted ORACLE source code. For example, we do not know if a reference validation mechanism (i.e., reference monitor) checks each subject's

and object's sensitivity label before mandatory access in granted or denied, or if some other scheme or procedure is used to match and compare subject and object labels.

HP-UX BLS enforces the MAC policy by making sure that the user process is cleared to access information from an object by comparing the sensitivity level of the process with the sensitivity level of the object. Therefore, this requirement is met in both the DBMS TCB and the OS TCB components.

5. MAC.5 Requirement

Trusted ORACLE supports the range, size, and type of label formats provided by HP-UX BLS [ORAC92b].

HP-UX BLS has been designed to support many different configurations of sensitivity labels, with a "virtually unlimited capacity for the number of classifications and categories."[HEWL92a] The maximum classification number is set to 16 by default; the maximum category number is set to 1024 by default. Both these defaults can be changed during system installation. [HEWL92a] Therefore, this requirement is met in both the DBMS TCB and the OS TCB components.

6. MAC.6 Requirement

Trusted ORACLE provides the read operation by granting its subjects the SELECT operation. Before a database user can SELECT from an object, such as a table or view (thus reading the object), his/her label (clearance) must dominate the label of the object. The MAC rules for reading an object state, "users can read objects at their label and below; users cannot read objects at labels that they do not dominate."[ORAC92a]

The Trusted ORACLE rules above use the term "dominate". Oracle defines dominate as a relationship between labels where one label dominates another if its classification is greater than or equal to that of the other label and its categories are a superset of the other's categories (all categories are represented). This is essentially the same definition as used in the TCSEC.

The HP-UX BLS MAC rules for reading an object are that a "subject's classification must be greater than are equal to the object's, and the subject's set of categories must include the object's."[HEWL92a] Therefore, this requirement is met in both the DBMS TCB and the OS TCB.

7. MAC.7 Requirement

Trusted ORACLE provides the write operation by granting its subjects the INSERT, UPDATE, or DELETE operation. Before a database user can perform one of these three operations to modify an object, such as a table or view (thus writing the object), his/her label (clearance) must match (equal) that of the label of the object. The MAC rules for writing an object states that "a user's label must match that of the object"[ORAC92a]. This is clearly a modification of the Bell-LaPadula security model which allows the subjects the ability to write to objects at higher labels. As in the Bell-LaPadula model, subjects are prevented from writing to objects at lower levels to prevent the possibility of unauthorized disclosure (i.e.,re-classifying information at a lower level). However, the WRITE, as defined in the TCSEC is met.

The HP-UX BLS MAC rules for writing an object are that "the object's classification must be equal to the subject's, and the object's set of categories must match the subject's."[HEWL92a] Therefore, this requirement is met in both the DBMS TCB and the OS TCB.

8. MAC.8 Requirement

In Trusted ORACLE it is mandated that the authentication of database users be conducted by the underlying operating system. This is done by specifying the IDENTIFIED EXTERNALLY parameter of the CREATE USER command whenever a new username is being created for the database.

HP-UX BLS authenticates by searching the Protected Password database for the user's ID clearance and encrypted password. The user, upon logging into the system must enter three items: login name, password, and sensitivity level.

The user can log on the system at any sensitivity level up to his/her clearance, which is the highest sensitivity level the user has been cleared for. (The clearance assigned to a user is determined by personnel policies, commensurate with the level of trustworthiness of the user; the Authentication Administrator sets up the user's clearance during system account creation.) In HP-UX BLS, the user's login process is tagged with the Login User ID (LUID). This indelible tag can never be changed or modified, not even by a superuser with all the system privileges. For example, even if a system administrator jumps from one user account to another (e.g., when using the *su* command), the LUID of the system administrator is inherited by the new processes spawned from the *su* program. This provides absolute accountability and always traces who did what by examination of the LUID. Therefore, this requirement is met in the OS TCB only.

9. MAC.9 Requirement

This requirement can be summarized to mean that system processes for users must be dominated by user's clearance as found in the system's I&A database. Based on available documentation, both the DBMS and the OS adhere to this requirement.

10. Additional MAC Comments

The user's LUID label (session level) is the sensitivity level that is used in Trusted ORACLE when subjects (processes) are created. No spawned process or new object created by the LUID process can exceed the sensitivity level of the LUID. The same is true when invoking special MAC privileges (WRITEDOWN, WRITEUP, READUP). When a database user connects to an Oracle database, he/she can connect at any sensitivity level up to his/her operating system clearance (I&A data).

With respect to the MAC privileges, if a database user is connected at UNCLASSIFIED and has the READUP privilege, then he or she can read higher levels of information, but only up to (and equal to) his/her overall system clearance. The READUP privilege violates the simple security property of the Bell-LaPadula model and is used as a means to prevent the user from having to log out of the system and then log back in at a

higher session level. The WRITEDOWN privilege is used for the same purpose (so the user does not have to log out and then in, to change his/her session level) but to write data. This privilege clearly violates the *-property of the Bell -LaPadula model and is utilized for administrators doing things such as a full database import. The WRITEUP privilege does not violate the Bell-LaPadula model, because writeups are allowed by definition in Bell-LaPadula, but not in the Trusted ORACLE MAC policy. Again, this privilege is most useful for administrators doing full database imports, but standard users can also use it as well. In summary, all MAC privileges granted to users must be dominated by the user's clearance. The DBA who grants the MAC privileges must have a session sensitivity level which dominates the user receiving these privileges.

The same can be said about HP-UX BLS. The special privilege, *allowmacaccess*, allows a process to reset its sensitivity label. This trusted process can only allow MAC access up to and including the clearance of the user's I&A data found in the Protected Password database. Therefore, this requirement is met in both the DBMS TCB and the OS TCB.

IX. INFORMIX ANALYSIS

This chapter discusses the analysis of Informix-OnLine/Secure 5.0 against the TCSEC requirements (as decomposed in Chapter VII) for labels and the mandatory access controls. The configuration that we analyzed is the one where Informix-OnLine/Secure utilizes the "raw device storage", thus circumventing most of the UNIX file system.

Again, we start by looking at the DBMS TCB component (i.e., database server software and user's manuals), then proceed to the operating system TCB. As discussed in Chapter VII, a requirement, as listed below, can be met in the DBMS TCB component, the operating system TCB component, both components, or it may not be meet or is not applicable. If user action is required to meet this requirement, then a "U" will be placed in the respective columns associated with the requirement. After each requirement, we determine where the requirement was meet, if in fact they were met at all. (See requirement summaries in Chapter VII.)

A. LABELS

1. LAB.1 Requirement

For an operating system user to gain access to Informix-OnLine/Secure, he/she must be added to a new operating system group called "ix_users." This requires the operating system administrator (OSA) to add this new group to the existing groups found in the HP-UX BLS implementation. In addition, the new database user's clearance must at least equal (i.e., dominate) the minimum clearance established for the Informix-OnLine/ Secure database (which is usually referred to as DATALO).

The database user's clearance (i.e., sensitivity label) is actually maintained within the OS tables (Protected Password database in HP-UX BLS). The Informix-OnLine/ Secure DBMS has no username tables which it actually maintains for user subjects.

Therefore, the OS component maintains all user subject labels of the system and an "OS" is given to HP-UX BLS component (and a NA to the DBMS component) in the summary tables in Chapter VII.

2. LAB.2 Requirement

All Informix-OnLine/Secure database objects and their labels are maintained within the *systables*, *sysprocedures* table, or a table's row table along with their labels. Each database defined (using the CREATE DATABASE SQL command) contains its own *systables*. The *systables* are in effect the system catalog (data dictionary) and are always established when a database is created. The objects found within the Informix-OnLine/ Secure database are database, table, row, blob, view, synonym, index, contraint, stored procedure. (This list is inclusive for database objects found in Informix-OnLine/Secure 5.0.) A hypothetical *systable* is shown in Figure 19, below. The table shows how each object is defined with its LABEL, type object, name and other attributes. The table object points to its respective row table, so that each row (in a table) and its label can be identified, as shown in Table 20, below.

TABLE 19: EXAMPLE OF A *SYSTABLE* IN INFORMIX-ONLINE/SECURE

LABEL	object type	object name	other attributes
10	database	Personnel	
10	table	Employee	Pointer to row table
100	view	CS_employ	
50	synonym	Private	
100	index	Quick	
50S	constraint	Excel	

Note that the LABEL values are in the tag format, not the human readable form, such a TOP SECRET or UNCLASSIFIED. For a user to return a human-readable label,

they must call the LABELTOSTRING function on the LABEL attribute of the respective table.

TABLE 20: EXAMPLE OF A ROW TABLE FOR A PARTICULAR TABLE OBJECT

LABEL	ROW_ID	Data fields
10	1	
50	2	
100	3	

Note that the LABEL and ROW_ID columns in both tables above, are invisible to standard users.

A hypothetical *sysprocedures* table is shown in Figure 21, below. The *sysprocedures* table is a special table separate from the *systables* and contains only the stored procedure objects defined on the database. Because the label of the row in the *sysprocedures* table is the same as the level of the procedure to which it refers, the result of a query on the LABEL attribute returns the security level of the procedure [INFO93c].

TABLE 21: EXAMPLE OF A *SYSPROCEDURE* TABLE IN INFORMIX-ONLINE/ SECURE

LABEL	procname	pointer
10	PROC1	
50	PROC2	
100	PROC3	

The storage objects seen by the HP-UX BLS operating system (i.e., files, directories, devices, IPC objects, symbolic links, named pipes, processes, printer queues, and ptys) are maintained within the OS. Files and directories are labeled individually, and

are labeled in such a way that the access classes increase as you go down the tree. The UNIX file system maintains all the files and directories within the OS TCB. All dynamic objects, such as printer queues, pipes and links, (and processes seen by HP-UX BLS) are also maintained in the OS TCB component.

Therefore, both components (the OS and the DBMS) are needed to maintain the object's labels of the system and a "B" is given to each component in the summary tables in Chapter VII.

3. LAB.i1 Requirement

If an object is to be a "public object" in Informix-OnLine/Secure, then it should be created at the lowest sensitivity level of data found in the database, which is DATALO. Additionally, the object hierarchy must be obeyed, which states that if a row within a table is public, then the table and the database must also be public as well.

The same argument is used in the objects maintained by the operating system. If files and directories are to be public, both the files and their parent directories must be labeled at System Low in HP-UX BLS.

Therefore, both components (the OS and the DBMS) are needed to maintain that certain objects be public within the system and a "B" is given to each component in the summary tables in Chapter VII.

4. LAB.3 Requirement

This decomposed requirement is identical to MAC.4. (See "MAC.4 Requirement" on page 133.)

5. LAB.4 Requirement

Standard users (not the DBSA) import single level data using the *dbimport* command. However, this data must be in ASCII format with no internal MAC labeling. All imported objects inherit the session level of the user's importing process.

Therefore, this requirement that the TCB shall request and receive the security level of the data, is accomplished directly by the authorized users of the system when they properly load tapes or floppy diskettes at the correctly labeled input device. This requirement is meet by the standard user, and we place a "U" in the DBMS column of the summary table, and a "NA" in the OS column.

6. LAB.5 Requirement

Standard users (not the DBSA) import unlabeled data using the *dbimport* command. The *dbimport* command is not audited, but other events associated with importing data (such as creating a new database, or locking of tables) are auditable. If a new database is not created, then this importing of non-labeled data may not be audited by Informix-OnLine/Secure. Therefore, this requirement for the auditing of non-labeled data will not have been met.

All actions performed by the DBSA and the DBSSO users are audited by default. There are no audit masks for DBSA and DBSSO users [INFO93b]. Only the DBSA can import data with OnLine/Secure internal MAC labeling from secondary storage media into an Informix-OnLine/Secure database. If the DBSA imports non-labeled data, then this requirement is met. However, if a standard user imports non-labeled data, then this action may not be audited by the system. Therefore, we place a "NM" in the DBMS column.

The HP-UX BLS operating system allows for the collection of audit data through the use of the Audit System Collection Mask. One of the audit capabilities of this system mask is the auditing of subsystem events. Since import/export of data is a subsystem event (i.e., Tape), the HP-UX BLS system is capable of auditing all events associated with the import of non-labeled data. Therefore, we place an "OS" in the HP-UX OS column of the summary tables for this requirement.

B. LABEL INTEGRITY

1. LI.1 Requirement

Since there is no separate login procedure for users using the Informix-OnLine/ Secure database, the subject's sensitivity levels and respective labels are maintained by the operating system administrator. The subject's sensitivity level (i.e., session level for the database) is determined when he/she logs into the operating system. This label is referred to as the session sensitivity level or simply session level.

The users of the system determine what sensitivity level an object is created at, and the TCB correctly maintains this sensitivity level with the object. For schema objects, we know that a label attribute is attached to each row of the *SYSTABLE* (See "LAB.2 Requirement" on page 118.) We cannot determine exactly how the subject label is used and maintained in the TCB, but we have determined that it is done, and give this requirement a "B"; both components enforce label integrity on subjects and objects.

2. LI.2 Requirement

Only the DBSA is capable of exporting labeled OnLine/Secure objects. (Standard users can import/export data using the LOAD and UNLOAD statements, or the *dbload*, *dbimport*, and *dbexport* utilities, but the object sensitivity labels are not preserved.) The sensitivity labels are written to the media in the tag representation (32 bit integer constant) of the label. (See LI.3 Requirement, below and EM.1 for more information.)

Therefore, this requirement is met by the DBMS component and a "D" is placed in the respective column and a "NA" in the OS column. (If the UNIX file system where utilized in Informix-OnLine/Secure in lieu of raw device storage, then an "OS" would be placed under the OS column and a "NM" under the DBMS column.)

3. LI.3 Requirement

The sensitivity labels are written to the media in the tag representation (32 bit integer constant) of the label. All *systables*, row tables, and *sysprocedures* tables are tagged

by row (under the LABEL attribute). These labels are unique to the specific implementation of the operating system; they are not necessarily the same or even understandable by other secure system implementations. Only the DBSA can export labeled data with OnLine/Secure internal MAC labeling to secondary storage media.

This requirement is met by the DBMS component and a "D" is placed in the respective column and a "NA" in the OS column.

C. EXPORTATION OF LABELED INFORMATION

1. EL.1 Requirement

The Informix-OnLine/Secure user's manual describes the desired sensitivity levels assigned to different devices as shown in Table 22:, below.

TABLE 22: SECURITY RANGES FOR DEVICES IN INFORMIX-ONLINE/ SECURE [INFO93B]

Device type	Minimum	Maximum
Terminal	Datalo	Datahi with groups IX_DATA, IX_DBSA, and IX_DBSSO (when defined)
Printer	Datalo	Datahi with groups IX_DATA, IX_DBSA, and IX_DBSSO (when defined)
Tape drives for DBSA use	Datahi, and IX_DATA	Same as minimum
Tape drives for standard users	Datalo	Datahi

However, the functions to designate each communication channel and I/O device are found within the underlying operating system, HP-UX BLS. The system administrator defines the security characteristics of each import/export device by placing the required information in the Device Assignment database. Every device that is to be used must have an entry in the Device Assignment database. Devices include terminals, line printers, and import/export devices such as tape drive systems. The operating system uses

this database to restrict data objects that have sensitivity levels outside the range of the devices specified, from either being sent or received through that device, thus preventing inappropriate data being disclosed. Every device is labeled specifically as either single or multilevel.[HEWL92a] Therefore, this requirement is met in the OS TCB component.

2. EL.2 Requirement

Informix-OnLine/Secure Trusted Facility Manual specifically requires that the DBSA seek the operating system documentation for instructions for assigning sensitivity levels to devices. OnLine/Secure does not provide any procedures for assigning sensitivity labels to I/O devices.

The operating system administrator, by invoking the *devasgif* command within the HP-UX BLS environment can change the parameters of the Device Assignment database. Only the system administrator with the proper kernel authorization and privileges can modify the Device Assignment database. The database contains information about all logical devices in the system, with each entry describing the devices characteristics (e.g., terminal, printer, removable media, such as a tape or floppy disk), whether its single-level or multilevel, or an import, export or both device.Therefore, this requirement is met in the OS TCB component.

3. EL.3 Requirement

Since the DBSA or the DBSSO in Informix-OnLine/Secure cannot change device designations, their actions, though always auditable, cannot capture the changing of device reassignments

All operating system administrator/operator actions are auditable if they are found in the Audit System Collection Mask [HEWL92a]. The actions performed by the System Administrator, including the use of the *devasgif* command to change the designation of a communication channel are auditable.Therefore, this requirement is met in the OS TCB component.

4. EL.4 Requirement

The HP-UX BLS maintains the I/O device labels in the Device Assignment database, a part of the operating system file structure. Any change to these security levels are found in the respective audit files or logs.Therefore, this requirement is met in the OS TCB component.

5. EL.5 Requirement

This requirement is the practically the same as EL.3 All operating system administrator/operator actions are auditable in they are found in the Audit System Collection Mask [HEWL92a]. The actions performed by the System Administrator, including the use of the *devasgif* command to change security levels of devices are auditable.Therefore, this requirement is met in the OS TCB component.

6. EL.i1 Requirement

Since all operating system administrator actions are auditable (if selected in the audit system collection mask), level changes of both single level and multilevel communications channels and I/O devices will be auditable. Though this interpretation states that multilevel communications channels and I/O devices are not required to be auditable, they will be in HP-UX BLS if the system administrator actions are selected in the Audit System Collection Mask because there is no granularity in the audit mechanism. (When the system administrator actions are audited, all actions associated with the system administrator are audited.)Therefore, this requirement is met in the OS TCB component.

D. EXPORTATION TO MULTILEVEL DEVICES

1. EM.1 Requirement

Only the DBSA is capable of exporting labeled Informix-OnLine/Secure objects. The DBSA can use five functions to export labeled data: the DB-MONITOR, the *tbtape -s*, *tbtape -a*, *tbtape -c* or the *tbunload* commands. The data is written to the media

retaining sensitivity labels in the tag format as stored in the Informix-OnLine/Secure database [INFO93b].

The *mltape* command within the HP-UX BLS operating system exports multilevel data out of the system.Therefore this requirement is meet by both the DBMS and the OS component of the TCB.

2. EM.2 Requirement

All the tasks allowed by the DBSA in Informix-OnLine/Secure to export multilevel data (as noted in EM.1 requirement above), write the data to the device (either a default device or a command line option) specified with the OnLine/Secure labels in their tag format. Therefore this requirement is met by the DBMS TCB component.

3. EM.i1 Requirement

All the tasks allowed by the DBSA in Informix-OnLine/Secure to export multilevel data (as noted in EM.1 requirement above), write the data to the media specified with the OnLine/Secure labels in their tag format. Standard users can export data using the UNLOAD statements, or the *dbexport* utility and the object sensitivity labels will not be preserved. The user is responsible to correctly label removable media if they transport unlabeled information out of the database. The label placed on the media must reflect the level of the classified information that the media contains. Therefore this requirement is considered a user requirement since the standard users of the system must properly label tapes and floppies with label that reflect that of the data.

4. EM.3 Requirement

The multilevel data exported by Informix-OnLine/Secure, is exported in its tag representation (i.e., machine-readable) format. If data is to be exported to a multilevel printer (human-readable from), then the HP-UX BLS printer utilities place the sensitivity label produced on the banner page of the printout. This label is the sensitivity level of the user executing the print command. (The user's sensitivity level is determined at the time

they log-on the system.) Therefore, depending on the export being made (either to tape or to printer), this requirement is met by both the OS and the DBMS TCB.

5. EM.4 Requirement

Informix-OnLine/Secure requires the use of the Informix-Star/Secure distributed client/server database product to enforce the database server's security policy to remote client workstations. The minimum requirement for a secure Class B1 configuration is MaxSix 1.0 networking software. However, client workstations can be running any network configuration, ranging from unlabeled to MaxSix 2.x, as long as the network security officer can configure the MaxSix network databases properly [INFO93f]. (For more information on MaxSix, See "EM.4 Requirement" on page 106.)

Additionally, the HP-UX BLS network transports security attributes with data and extends a host's access controls to the network subsystem so that a host can make access decisions using local policies as data traverses the network between communicating processes [HEWL92c].Therefore this requirement is meet by the OS TCB component.

6. EM.5 Requirement

This requirement is met by the MaxSix MLS network protocol. See EM.4 requirement, above. Therefore this requirement is met by the OS TCB component.

E. EXPORTATION TO SINGLE-LEVEL DEVICES

1. ES.1 Requirement

The Informix-OnLine/Secure DBMS relies upon the operating system administrator to set the device security attributes in coordination with the DBSA [INFO93b].

The HP-UX BLS operating system designates and maintains (through the operating system administrator) a device in the system as either single-level or multilevel. A single level device does not store labels with the files that it outputs. However, HP-UX BLS does specify a default sensitivity level for a single level device. This means that if a

single level device is labeled, then only data at that level is exported through that device. Of course no labels will be associated with the exported files. Therefore, this requirement is met by the OS TCB component only.

2. ES.2 Requirement

Standard Informix-OnLine/Secure users can import single-level data using the *dbload* or the *dbimport* utility or the SQL LOAD command. (However, the *dbimport* is only good if the file being imported was created with the *dbexport* utility.) The *dbload* utility transfers data from one or more ASCII files into one or more existing tables within the database. All data imported into the database is at the level of the user. Therefore, for the DBMS component of the TCB, we have labeled this requirement as "U", for user responsibility.

The *tar* and *cpio* programs of HP-UX BLS have been modified to import data from single-level media.These programs perform the appropriate checks against the Device Assignment database to ensure that the device used for import is in fact specified as single-level. Therefore, this requirement is met by the OS TCB and we labeled the OS column with an "OS".

3. ES.3 Requirement

Only standard users export non-labeled (i.e, single-level) data from Informix-OnLine/Secure using the *dbexport* utility or the UNLOAD SQL command. The dbexport unloads rows from a database into ACSII files. The information exported is all the information that is dominated by the user's session sensitivity level and only advisory labels (not real labels) can be requested The *dbexport* utility can export information directly to a file or a specified device, such as a tape device, but it is assumed that it must rely on the HP-UX BLS Device Assignment database to determine if the device is single-level.

If a file, created using the *dbexport* utility is later exported out of the system the *tar* and *cpio* programs of HP-UX BLS can be called to export data to single-level medium. These programs perform the appropriate checks against the Device Assignment database

to ensure that the device used for export is in fact specified as single-level. The Informix-OnLine/Secure *dbexport* utility must confer with the HP-UX BLS Device Assignment database before exporting single-level data. Therefore, this requirement is can be met by both the OS TCB component.

F. LABELING HUMAN-READABLE OUTPUT

Informix-OnLine/Secure provides the DB-Access utility which allows a database user to redirect the results of an SQL query (e.g., SELECT) from the screen to a printer, system file, or a program. By selecting the "Printer" option on the OUTPUT menu, the DB-Access utility sends the results of a query to the default printer. [INFO93e] The actual printing of human-readable output is handled by the HP-UX BLS operating system.

1. HRO.1 Requirement

The operating system administrator only indirectly specifies the printable label names associated with exported security labels. Labels names are set-up when the sensitivity classifications and categories are defined for the system. Whenever objects are created (either database objects or operating system objects), they are labeled with the creating process's sensitivity level.

HP-UX BLS prints the sensitivity label of the process producing the output on the banner page of the printout. This is the sensitivity label that was typed when the user logged into the system. Therefore, this requirement is met by the OS TCB component.

2. HRO.2 Requirement

The HP-UX BLS prints the sensitivity label of the process producing the output on the banner page of the printout. It usually appears in the same location on the banner page as the print-job detail. As stated in HRO.1, this is the sensitivity label that was typed when the user logged into the system.

There is no mention of marking the end of all human-readable paged, hardcopy output with a trailer page. However, the last page printed (a body page) will have a human-

readable sensitivity labels printed on the top and bottom of the page. Therefore, this requirement is met by the OS TCB component.

3. HRO.3 Requirement

When the information is printed on the printer by the HP-UX BLS operating system, the banner page includes the sensitivity level of the process and each internal page includes the sensitivity level of the file that appears on that page. Internal pages (body pages), are labeled with the highest sensitivity level of the information that is printed on the page.

Top and bottom labeling is characteristic only of the body pages of the printout. The banner page only prints the classification one time, usually near the print-job detail. There is no mistaking a banner page with a body page in HP-UX BLS. Therefore, this requirement is met by the OS TCB component.

4. HRO.4 Requirement

This requirement does not appear to be applicable in HP-UX BLS operating system. Labeling is supported on the line printer only. Labels are not supported on laser printers or plotters [HEWL92c]. In addition, labeling is not supported when the output is assigned a "landscape" orientation (i.e., the output is printed horizontally)[HEWL92c]. However, since HP-UX BLS does not support these printing options (it only prints line printer text) this requirement is not applicable to the OS TCB or the Trusted Oracle TCB components.

5. HRO.5 Requirement

HP-UX BLS provides two secondary line printer subsystem options which can override the labeling and filtering capabilities of normal printing. The *label* authorization allows the use of the *-l* option to the *lp* command to suppress the labels applied to the internal pages of a printout. The *filter* authorization allows the *-f* option to the *lp* command,

which removes the filtering done on printed output. Both of these options are audited by the system. [HEWL92a]

As a side note, both the special options to the *lp* command suppress the internal printout labeling features of HP-UX BLS. The assumption is that the banner page is still printed. Therefore, this requirement is met by the OS TCB component.

G. MANDATORY ACCESS CONTROL

1. MAC.1 Requirement

Subjects in Informix-OnLine/Secure are operating system processes that use On-Line/Secure. The DBMS server mandates that every access to every piece of data (i.e., object) by all users (i.e., subjects) be checked to see if access is permissible, based on the sensitivity label of the data and clearance of the user. Informix-OnLine/Secure uses the same set of sensitivity labels that are available in HP-UX BLS. Therefore, there is no problem with comparison of subject and object labels when the DBMS assigns labels to objects based on the subject's sensitivity level. (Specific access rules will be discussed in MAC.6 and MAC.7 requirements.)

The HP-UX BLS TCB component maintains sensitivity labels on all subjects (active entities in the system, such as processes). When a user logs into the system, a login user ID (LUID) is attached to the user's login process. The LUID points to the Protected Password database which contains the clearance level for that particular subject. (The clearance level is the highest sensitivity label allowed for that subject's process). All processes spawned from this LUID process contain this same LUID with its associated clearance level. From these subject processes' labels a mandatory access control policy, based on the Bell-LaPadula security model is enforced. (See the MAC.6 and MAC.7 requirements for details below.) Therefore, this requirement is met in both the DBMS TCB and the OS TCB components.

2. MAC.2 Requirement

Informix-OnLine/Secure has several object types which are different from the operating system objects. These DBMS objects are databases, tables, rows, blobs, views, synonyms, indexes, contraints, and stored procedures. All database objects have labels associated with them. (See the LAB.2 requirement for details.) Again, as in the requirement above (MAC.1) the DBMS server mandates that every access to every piece of data (i.e., object) by all users (i.e., subjects) be checked to see if access is permissible, based on the sensitivity label of the data and clearance of the user. As stated in MAC.1 above, Informix-OnLine/Secure uses the same set of sensitivity labels that are available in HP-UX BLS. (Specific access rules will be discussed in MAC.6 and MAC.7 requirements.)

The HP-UX BLS TCB component maintains sensitivity labels on all objects controlled by the operating system. These objects include regular files, inter-process communication (IPC) objects, directories, special files, pipes, processes, and symbolic links. When objects are created, the HP-UX BLS system attaches security attributes to the objects. For example, when a file is created, the full pathname of the file, the file owner and group, the file mode and type, the sensitivity level, the potential and granted privilege sets, and the access control lists are stored in the File Control database. Everytime a process attempts to access the file, it must search the File Control database, check, and pass each parameter before access is granted. The MAC policy for reading and writing these objects is discussed in depth in the MAC.6 and MAC.7 requirements below. Therefore, this requirement is met in both the DBMS TCB and the OS TCB components.

3. MAC.3 Requirement

The subject (application process) receives its label when the user logs into the operating system. This subject label is referred to as the session sensitivity label or session label. Objects receive their sensitivity label when they are created. Subject and object sensitivity labels in Informix-OnLine/Secure are composed of the following two components: a hierarchial component, such as UNCLASSIFIED, SECRET, TOP

SECRET, called an access level, and zero or more categories, such as CRYPTO, NATO, and PROPRIETARY[INFO93c]. Informix-OnLine/Secure uses the same set of sensitivity labels that are available in HP-UX BLS. (The DBSA sets up labels in the DBMS to equal those in the OS.) Therefore, there is no problem with comparison (i.e., label equality, label dominance, etc.) of subject and object labels when the DBMS assigns labels to objects based on the subject's sensitivity level.

The HP-UX BLS system specifically supports the US DOD method of classifying information according to hierarchial classification levels and non-hierarchial categories. Therefore, this requirement is met in both the DBMS TCB and the OS TCB components.

4. MAC.4 Requirement

Subjects can access objects in Informix-OnLine/Secure by comparing sensitivity labels (in their integer tag format) of objects and subjects.

HP-UX BLS enforces the MAC policy by making sure that the user process is cleared to access information from an object by comparing the sensitivity level of the process with the sensitivity level of the object. Therefore, this requirement is met in both the DBMS TCB and the OS TCB components.

5. MAC.5 Requirement

Informix-OnLine/Secure uses the same set of sensitivity labels that are available in the operating system [INFO93c]. (See below.)

HP-UX BLS has been designed to support many different configurations of sensitivity labels, with a "virtually unlimited capacity for the number of classifications and categories."[HEWL92a] The maximum classification number is set to 16 by default; the maximum category number is set to 1024 by default. Both these defaults can be changed during system installation. [HEWL92a] Therefore, this requirement is met in both the DBMS TCB and the OS TCB components.

6. MAC.6 Requirement

In Informix-OnLine/Secure, subjects can read only objects that they dominate. A sensitivity label is said to dominate another sensitivity label when the access level (i.e., SECRET) of the first sensitivity label is greater than or equal to that of the second sensitivity label and the set of categories of the first sensitivity label is a superset of or equal to the set of categories of the second sensitivity label [INFO93c].

The HP-UX BLS MAC rules for reading an object are that a "subject's classification must be greater than or equal to the object's, and the subject's set of categories must include the object's."[HEWL92a] Therefore, this requirement is met in both the DBMS TCB and the OS TCB components.

7. MAC.7 Requirement

In Informix-OnLine/Secure, subjects can write only objects at their security level [INFO93c]. This is clearly a modification of the Bell-LaPadula security model (BLM) which allows the subjects the ability to write to objects at higher labels. As in the BLM, subjects are prevented from writing to objects at lower levels to prevent the possibility of re-classifying information at a lower level.

The HP-UX BLS MAC rules for writing an object are that "the object's classification must be equal to the subject's, and the object's set of categories must match the subject's."[HEWL92a] Therefore, this requirement is met in both the DBMS TCB and the OS TCB components.

8. MAC.8 Requirement

All I&A functions are handled exclusively by the underlying operating system in Informix-OnLine/Secure; there are no options on where I&A can be done.

HP-UX BLS authenticates by searching the Protected Password database for the user's ID clearance and encrypted password. The user, upon logging into the system must enter three items: login name, password, and sensitivity level.

The user can log on the system at any sensitivity level up to his/her clearance, which is the highest sensitivity level the user has been cleared for. (This clearance is established by the security policy outside the automated system environment and the Authentication Administrator sets it up during system account creation). In HP-UX BLS, the user's login process is tagged with the Login User ID (LUID). This indelible tag can never be changed or modified, not even by a superuser with all the system privileges. For example, even if a system administrator jumps from one user account to another (e.g., when using the su command), the LUID is still inherited by the new processes spawned from changes accounts. This provides absolute accountability and always traces who did what by examination of the LUID. Therefore, this requirement is met in the OS TCB only.

9. MAC.9 Requirement

This requirement can be summarized to mean that system processes for users must be dominated by user's clearance as found in the system's I&A database. Based on available documentation, both the DBMS and the OS adhere to this requirement.

10. Additional MAC Comments

Additionally, in Informix-OnLine/Secure, trusted subjects or processes are created when invoking discrete privileges (PRIV_CANSETLEVEL and PRIV_CANSETIDENTITY). The PRIV_CANSETLEVEL allows a user to alter the session sensitivity level of the current session by invoking the SET SESSION LEVEL statement. The user can operate only at sensitivity levels that are dominated by the level of his/her original login session. The SET SESSION LEVEL allows the user, (by creating temporary tables and then changing session level), to change the object levels. This is a violation of the tranquility property of Bell-Lapadula model.

The PRIV_CANSETIDENTITY discrete privilege allows users to circumvent the DAC protection for database objects by adopting the user name of any Informix-OnLine/Secure nonadministrative user. The user's login ID and sensitivity level still determine what level of data the user holding this privilege can access.

The same can be said about HP-UX BLS. The special privilege, *allowmacaccess*, allows a process to reset its sensitivity label. This trusted process can only allow MAC access up to and including the clearance of the user's ID as found in the Protected Password database. Therefore, this requirement is met in both the OS TCB and the DBMS TCB components.

X. COMPARISON OF TRUSTED ORACLE AND INFORMIX

In this chapter we will list both the similarities and the differences of Trusted ORACLE 7 and Informix-OnLine/Secure 5.0 in the context of the TCSEC requirements examined in the two previous chapters.

A. LABELS

1. LAB.1 Requirement

Trusted ORACLE maintains an ALL_USERS table for storing the database (user) subjects, by keeping a user's name, clearance, and security profile. Informix-OnLine/Secure has no such comparable table, and exclusively uses the HP-UX BLS operating system Protected Password database to maintain its database users and their security clearances. Informix requires that three new groups be added to the HP-UX BLS groups: ix_data, ix_dbsso, and ix_dbsa.

2. LAB.2 Requirement

Trusted ORACLE 7.0 and Informix-OnLine/Secure 5.0 both maintain their data dictionaries by defining objects as rows in a system table. Each row (which defines a specific object) is labeled with the sensitivity level of the subject that created it and thus is the objects label. The objects in each system are similar in nature, as shown in Table 23, below. An "*" placed in a column means that specific product does not have a comparable object in the other product.

TABLE 23: COMPARISON OF DATABASE OBJECTS

Trusted ORACLE 7	Informix-OnLine/ Secure
Database	Database
Tables	Tables
* (NOTE 1)	Rows
Clusters	*
Indexes	Indexes
Views	Views
Sequences	*
* (NOTE 2)	Synonyms
Rollback Segments	*
* (NOTE) 3	Blobs
Stored Procedures and Functions	Stored Procedures
Triggers	Constraints
Packages	*

NOTE 1: The *Trusted ORACLE Administrator's Guide* [ORAC92a] does not describe "rows" as objects (though the Technical Overview [ALLE94] does call rows database objects), while Informix-OnLine/Secure specifically does call rows objects [INFO93c].

NOTE 2: Synonym objects are supported in Trusted ORACLE OS MAC mode.

NOTE 3: Blobs in Informix-OnLine/Secure are Binary Large Objects. Blobs are data storage objects that effectively have no maximum size (theoretically, as large as 31^2 bytes). Informix-OnLine supports two blob data types: TEXT for string ASCII data and BYTE for storing any type of binary data. To our knowledge, there is not comparable object in Trusted ORACLE.

3. LAB.i1 Requirement

There are no significant differences between Trusted ORACLE 7.0 and Informix-OnLine/Secure 5.0 with respect to this requirement.

4. LAB.3 Requirement

There are no significant differences between Trusted ORACLE 7.0 and Informix-OnLine/Secure 5.0 with respect to this requirement. Both database server systems use labeled objects and subjects with an extended version of the Bell-LaPadula model to control mandatory access between subjects to objects within the system.

5. LAB.4 Requirement

Trusted ORACLE uses the Import utility to import a single level (or multilevel) OS file into the database. Informix utilizes the *dbimport* command to do this function. Users are responsible to ensure that their session levels match the sensitivity of the data being imported.

6. LAB.5 Requirement

When importing OS files into their respective databases, neither Oracle nor Informix audits the import utilities or commands. When the HP-UX BLS OS imports files into the system (from outside the system), these actions are audited.

B. LABEL INTEGRITY

1. LI.1 Requirement

Trusted ORACLE stores username accounts and schema objects as rows in a system table within the data dictionary and sensitivity labels are associated with these rows. Informix-OnLine/Secure stores only schema objects, as rows in a system table and labels each row with the objects sensitivity label. Label integrity is maintained by the TCB.

2. LI.2 Requirement

Standard users in Trusted ORACLE can export single level objects from the database, but labels are not exported when a single level export is conducted. Only users with special MAC privileges can export multilevel data, and labels are then maintained on the exported data as either a MLSLABEL(4 byte tag) or RAW MLSLABEL (OS label) data type. Informix allows only the DBSA to export multilevel data records out of the database (labels are exported in their tag format of 4 bytes); standard users can export single level data only.

3. LI.3 Requirement

Only multilevel exports in both Trusted ORACLE and Informix-OnLine/Secure associated labels with data. (See LI.2 above.)

C. EXPORTATION OF LABELED INFORMATION

All designation of communication channels and I/O devices is handled by the operating system, HP-UX BLS. The auditing of any changes to these communication channels and devices is likewise handled by HP-UX BLS. Therefore, there are no substantial differences between Trusted ORACLE and Informix-OnLine/Secure with respect to the requirements El.1 - El.5.

D. EXPORTATION TO MULTILEVEL DEVICES

1. EM.1 Requirement

There are no substantial differences between Oracle and Informix with respect to this requirement.

2. EM.2 Requirement

Trusted ORACLE requires the use of the HP-UX BLS command *mltape*, to send a multilevel file to an I/O device. Informix-OnLine/Secure requires the DBSA, using the

tbunload command, to send multilevel data directly to an output device, using the *-t* device option.

3. EM.i1 Requirement

There are no substantial differences between Oracle and Informix with respect to this requirement.

4. EM.3 Requirement

Trusted ORACLE sends labeled objects to I/O devices with their labels as either a MLSLABEL or RAW MLSLABEL data types. Informix sends labeled objects with labels as 4-byte integer tags.

5. EM.4 Requirement and EM.5 Requirement

The MaxSix secure networking package is required for Trusted ORACLE and Informix-OnLine/Secure. We believe that both products meet this requirement by utilizing the MaxSix protocol.

E. EXPORTATION TO SINGLE-LEVEL DEVICES

There are no substantial differences between Trusted ORACLE and Informix with respect to the requirements ES.1 - ES.3.

F. LABELING OF HUMAN-READABLE OUTPUT

All requirements decomposed under this heading are met in the same way by both Trusted ORACLE and Informix-OnLine/Secure, because both DBMSs rely on the underlying operating system, HP-UX BLS exclusively to print human-readable output to the line printers. We found no mention that sensitivity labels are printed to the terminal screens.

G. MANDATORY ACCESS CONTROLS

With respect to the decomposed requirements, MAC.1 - MAC.9, there are no substantial differences between Trusted ORACLE and Informix-OnLine/Secure. The

MAC policy enforced by each implementation is an extension of the Bell-LaPadula model, with the extension being that writeup of data is not allowed by default.

The one exception between the products is with respect to MAC.3, how labels are assigned to subjects and objects. The label format used in Oracle contains four components, as shown in Chapter VIII.(see Figure 22 on page 112) Apparently, Informix labels only consist of one component, the sensitivity component, and no integrity or other components are built in. It should be noted, again, that since HP-UX BLS does not support integrity components, the Oracle integrity component of the label is not used in this configuration.

H. ADDITIONAL MAC COMMENTS

The table below shows how the generic functions of writedown, readup, and writeup, are accomplished in the three products we analyzed.:

TABLE 24: COMPARISON OF SPECIAL PRIVILEGES

Privilege Description	HP-UX BLS	Trusted ORACLE	Informix-OnLine/ Secure
writedown (Trusted subject of BLP)	allowmacaccess, downgrade	WRITEDOWN	PRIV_CANSETLEVEL
readup (Violates simple security property of BLP)	allowmacaccess	READUP	PRIV_CANSETLEVEL
writeup (Allowed in BLP)	writeupclearance, writeupsyshi, allowmacaccess	WRITEUP	PRIV_CANSETLEVEL

The writedown privilege allows a higher level subject to write to a lower level object and requires a trusted subject in the Bell-LaPadula model (BLP). Each product allows writedowns, as indicated by Table 24. The readup function is a violation of the simple security property of the BLP and is allowed in all three products. Lastly, the writeup

privilege, which is allowed in the BLP model (but not in the extended-BLP which is the MAC model used in the three products), can be accomplished in the products by granting the appropriate special privileges, as indicated in Table 24.

The PRIV_CANSETLEVEL discrete privilege in Informix-OnLine/Secure, allows a database user to toggle back and forth between session levels without logging out and logging back into the system. Trusted ORACLE can accomplish this function by granting a user the MAC privileges (i.e., WRITEDOWN, READUP, WRITEUP).

The PRIV_CANSETIDENTITY in Informix-OnLine/Secure (not shown in Table 24 because, to our knowledge, Trusted ORACLE has no equivalent privilege) allows a database user to assume the identity of another database user, thus having access to that database user's owned objects. This is a way for Informix to bypass discretionary access controls, without becoming a special user, such as the DBSA or the DBSSO. Additionally, we do not know which privilege in the underlying operating system, HP-UX BLS, coincides with the PRIV_CANSETIDENTITY Informix privilege. The only user in the operating system, who can bypass discretionary access controls on unowned objects, is usually the *root* account (i.e., superuser). Therefore, we can only assume, that when a database user is granted the PRIV_CANSETIDENTITY, they have a superuser account.

XI. CONCLUSIONS AND RECOMMENDATIONS

This chapter will give overall findings from our research and make recommendations for a number of items which we feel can be improved upon. We conclude with some recommendations for future research in this area of DBMS security evaluation and analysis.

A. SUMMARY

The security of information within computer systems is a major issue for system automation professionals of the 1990's and beyond. The disclosure of sensitive information has plagued system administrators for many years, and today, with the advent of interactive network computing and the "information superhighway", information will have to be protected more than ever.

Class B1 (and higher) relational database management systems (RDBMS) are an inherent part of the solution for secure interactive network computing and the information superhighway. Our analysis of two leading RDBMSs, has demonstrated a feasible approach for system automation professionals to analyze and evaluate the merits of a multilevel secure database management system. Given only public documentation, an information systems professional can analyze the security features of a new product (before the release of the official evaluation results from the NSA, if in fact it is to be evaluated and rated by NSA at all) More importantly, this analysis and examination of the product, will give the administrator a much greater understanding of the operational merits of the system before a decision is made to even purchase the product.

1. Oracle Summary

The Trusted ORACLE 7 database server relies heavily upon the operating system on which it is placed. Trusted ORACLE's MAC policy is an extended version of

the Bell-LaPadula security model (i.e., subjects writing to objects must have equal sensitivity levels). Objects are stored in data files which can be stored in raw devices of the system controlled by the Oracle database server.

Trusted ORACLE does employ trusted subjects when a user executes the MAC privilege WRITEDOWN. The WRITEDOWN MAC privilege clearly violates the Bell-Lapadula model (*-property), and is truly a trusted subject as defined by [GASS88]. The WRITEDOWN privilege is utilized when conducting a full database import and a high-level (i.e, system high) process is allowed to write data to the database at its actual label, which can be lower than system high level.

The READUP MAC privilege of Trusted ORACLE is used by database users to read objects at higher levels than their session level. (This privilege violates the simple security property of the Bell-LaPadula model.) However, this privilege is limited to reading objects with a sensitivity label dominated by the overall clearance of the user, as defined in the operating system's Protected Password database. The READUP privilege only overrides the user's session sensitivity level, not the user's system clearance level.

In addition, the CREATE TABLE AS command allows a standard user with no MAC privileges to change his/her table(s) from one sensitivity label to another (up to their DBMS sensitivity label) [ORAC92a].This is actually done by creating a new table (with all the same attributes as the old table) at a new sensitivity level, and then copying all data from the old table into the new table. The new table's definition is the exact same as the old table, just with a new sensitivity label. The old table is then dropped from the database.

Any user in Trusted ORACLE can change the labels of his/her rows within their owned tables at any time. This raises a serious concern about the persistence of row and table object labels. The Bell-LaPadula Model implies the tranquility constraint that object access classes cannot change; object labels must remain unchanged through their lifetime. Trusted ORACLE circumvents this tranquility restriction, carefully (and apparently purposely), by not defining rows as objects. Instead, they call the changing of a row's ROWLABEL pseudo-column, the "reclassifying of data." [ORAC92a] To reclassify data

(i.e., rows in a table), a user must have MAC privileges (i.e, WRITEDOWN, READUP, WRITEUP). A standard user, with privileges, can change all labels in his/her own tables at anytime. Standard users cannot change labels of objects they do not have DAC permissions to, or if their clearance does not dominate the object's sensitivity label.

2. Informix Summary

Informix-OnLine/Secure, in the raw devices configuration, is capable of handling some database imports and exports directly, bypassing the HP-UX BLS file system.

Informix-OnLine/Secure also employs trusted subjects. A normal database user, with the appropriate discrete privileges can replace the sensitivity labels of objects (i.e., rows). If a user has the PRIV_CANSETLEVEL discrete privilege, they can execute the SET SESSION LEVEL statement and reclassify data labels. What this means is that a user with a system high clearance can change the labels of all the rows in all tables that they have access to. Additionally, if a user also has the PRIV_CANSETIDENTITY discrete privilege, they can change the labels of other nonadministrative user's tables as well. This would allow a standard user the ability to change all user row labels (given that he/she has system high clearance) in the database.

Changing labels on database, table, and row objects can be done by the DBSSO utilizing the Secure Administrator Front End. The only restrictions are the range of labels that can be used in the database server (i.e., Datahi and Datalo), the object hierarchy between rows, tables, and databases must be maintained, semantics of unique columns must stay the same, and some locking restrictions apply [INFO93b]. Standard users, as stated above, can change sensitivity labels if granted certain discrete privileges. This allows users with these privileges to change labels, at their discretion, when they decide to reclassify data.

3. Limitations of Research

A thorough, technical analysis was not possible with only public documentation. We acquired the following types of documentation for our research:

- Technical Overviews and Briefs
- Trusted Facility Manuals
- Administrator User's Guides
- Security Features User's Guides
- other public documents

These types of documents are insufficient to determine if products can meet the TCSEC requirements. These types of manuals and guides do not focus on the security features as they relate to the TCSEC and there is little or no discussion on the design of the system's security mechanisms. (The organization of the documents do not coincide, structurally with the TCSEC.)

B. RECOMMENDATIONS

1. MAC Downgrade Policy

All system administrators of a Trusted ORACLE or an Informix-OnLine/Secure database should develop a security policy for downgrading object labels. We recommend that only the system security officer in conjunction with the DBA be allowed to make object level changes, following a security policy for reclassifying data.(Separation of privilege would require both administrators to agree on object label changes before an actual change can be made in the system.) Specific rules with respect to this policy, cannot be defined beforehand, but a generic policy addressing the labeling and downgrading of objects should be established. This will allow users some notion of how to label objects and how to reclassify objects once they are defined. All reclassifying of data should be audited by default with no override capabilities.

2. Method Reformulation/Evolution

The method used in this research is in its infancy. To our knowledge, their is no comprehensive handbook available for system automation professionals to analyze a DBMS product for security features. The *INFOSEC Handbook: An Information Systems Security Reference Guide* [ARCA93] is the best comprehensive book we found to supplement the Rainbow series books with product analysis. This method of analysis can be extended to encompass all the requirements at a particular TCSEC class, not just the labels and mandatory access control requirements we chose to analyze.

3. Documentation Improvements

Mapping the TCSEC requirements to public documents, such as the ones we utilized in this research is not easy. The documents lack references to the TCSEC requirements and are not structured to make mapping easy. These DBMS implementations are designed to meet the Class B1 assurance level, yet their respective documentation is not structured to address security related items in the ordering presented in the TCSEC. (Informix has indicated plans to develop Class B2 and Class B3 versions of Informix-OnLine/Secure.) Our method showed that many requirements were hard to identify in the public documents available.

The downgrading of labeled objects and the security policy which dictates how, why, and by whom labeled objects can be downgraded, should be noted in documentation for each respective product. Neither Trusted ORACLE nor Informix-OnLine/Secure specified a generic security policy for downgrading objects. Any standard users, with the appropriate privileges can change the labels on some labeled objects (i.e., rows).

During the course of our examination, we did not find the TDI to be of great utility. The TDI does contain some interesting material on TCB subsetting and the evaluations of TCB subsets, but no real substantive interpretations for DBMSs exist in the TDI which was of use in our analysis.

C. FUTURE RESEARCH

Future research in this area could lead to a comprehensive evaluation handbook for the analysis of DBMSs. Upon the release of the Final Evaluation Reports by the NCSC for Trusted ORACLE and Informix-OnLine/Secure, study can continue to correlate the findings by NCSC with the findings of this research. Our analysis method can be expanded to cover all the TCSEC requirements, not just the MAC and labeling requirements.

A number of research questions have been posited over the years concerning the evaluation of software products which are "layered" or placed on top of trusted (previously evaluated) operating systems, such as DBMSs. Such questions are:

- Can a DBMS be evaluated like an operating system? An operating system manages resources, whereas a DBMS manages information.What is the relationship?
- Are TCB subset architectures, in the context of DBMSs, amenable to incremental evaluation (i.e., evaluating the application only without evaluating the underlying TCB subset [typically the operating system], which has already been successfully evaluated).[CHOK92] The basis for this question is the hope that the underlying operating system would not have to be reevaluated in order to evaluate the DBMS software product.
- Can or should the TCSEC, which was written initially for operating systems and their evaluations, be used for the evaluations of other systems, such as DBMS? Some authors have argued, that different criteria (criteria other than the TCSEC) should be developed for DBMS products. [GRAU90]

Future research in these areas can help determine the answers to these questions.

LIST OF REFERENCES

[ALLE94] Allen, Richard and Ehrsam, Tim, *Trusted ORACLE7 Technical Overview*, Oracle Corporation, 1994.

[ATKI94] Atkinson, Ran, Naval Research Lab, Washington, DC, Personal conversation via e-mail, 29 Aug 1994.

[AMOR94] Amoroso, Edward, *Fundamentals of Computer Security Technology*, PTR Prentice Hall, Englewood Cliffs, NJ, 1994.

[ANDE72] Anderson, J.P., *Computer Security Technology Planning Study*, ESD-TR-73-51, Vol I, AD-758 206, ESD/AFSC, Hanscom AFB, MA., October 1972.

[ARCA93] ARCA, INFOSEC Handbook: An Information Systems Security Reference Guide, 2nd Edition, ARCA, San Jose, CA 1993.

[BELL73] Bell, D., and LaPadula, L. "Secure Computer Systems: Mathematical Foundations and Model." *MITRE Report MTR 2547*, Vol. 2, Nov 1973.

[CAMP94] Campbell, Debbie, CDR, US Navy, Chief, Standards and Test Division, NCSC, Ft Meade, MD., Personal conversation via e-mail, 31 May 1994.

[CHOK92] Chokhani, Santosh, "Products Evaluation", Communications of the ACM, Vol 35, No 7, July 1992.

[DASH94] Dasher, Don, Oracle Corp., Personal conversation via e-mail, 8 Jun 1994.

[DATA94a] Datapro, *Datapro Computer Systems Series: Systems*, Datapro Information Services Group, McGraw-Hill Inc., Delran, NJ, February 1994.

[DATA94b] Datapro, *Datapro Computer Systems Series: Software*, Datapro Information Services Group, McGraw-Hill Inc., Delran, NJ, January 1994.

[DATE90] Date, C.J., *An Introduction to Database Systems*, Volume I, Addision-Wesley, 1990.

[DENN88] Denning, D.E., Lunt, T.F., Schell, R. R., Shockley, W.R., Heckman, M., *The SeaView Security Model*, IEEE, 1988.

[DOD85] Department of Defense, *Trusted Computer System Evaluation Criteria*, DOD 5200.28-STD, December 1985.

[DOWN94] Downs, Deborah, Chief Evaluator, Aerospace Corp., Los Angeles, CA.,Personal conversation via e-mail, 3 May 1994.

[DOYL91] Doyle, Sean, *An Introduction: Trusted ORACLE RDBMS*, Oracle Corporation, Redwood City, CA, February 1991.

[EDGE93] Edge Publishing, *EDGE: Workgroup Computing Report*, Version 4, Number 163, Edge Publishing Inc., July 5, 1993.

[EHRS91] Ehrsam, Tim, and Doyle, Sean, *ORACLE and Secure Systems: Questions and Answers*, Oracle Corporation, Redwood City, CA, October 1991.

[ENDO93] Endoso, Joyce, "DOD wants to Dump Mil Specs and Use More COTS Products", *Government Computer News*, Volume 12, Number 13, Page 3, June 21, 1993.

[GALL94] Gallagher, Sean, "Trusted ORACLE7 database is on brink of B1 security rating," *Government Computer News*, Volume 13, Number 4, February 21, 1994.

[GASS88] Gasser, Morrie, *Building a Secure Computer System*, Van Nostrand Reinhold, 1988.

[GRAU90] Graubart, Richard, "Comparing DBMS and Operating System Security Requirements: The Need for a Separate DBMS Security Criteria," *Database Security, III: Status and Prospects*, North-Holland, 1990.

[HALE94] Hale, Mike, Interview with Mike Hale at NCSC, Ft Meade, MD., 5 April 1994.

[HEWL92a] Hewlett-Packard, *HP-UX B-Level Security Trusted Facility Manual*, Edition 1, HP 9000 Computers, Hewlett-Packard Company, Palo Alto, CA, February 1992.

[HEWL92b] Hewlett-Packard, *HP-UX B-Level Security User's Guide*, Edition 1, HP 9000 Computers, Hewlett-Packard Company, Palo Alto, CA, February 1992

[HEWL92c] Hewlett-Packard, *B1 Networking Security Features User's Guide*, First Edition, HP 9000 Computers, Hewlett-Packard Company, Palo Alto, CA, January 1992.

[HINK90] Hinke, Thomas H., "DBMS Trusted Computing Base Taxonomy," *Database Security, III: Status and Prospects*, North-Holland, 1990.

[INFO93a] Informix, *Informix-OnLine/Secure Version 5.0 Technical Brief*, Informix Software, Inc., Menlo Park, CA, 1993.

[INFO93b] Informix, *Informix-OnLine/Secure Trusted Facility Manual Version 5.0*, Informix Software, Inc., Menlo Park, CA, April 1993.

[INFO93c] Informix, *Informix-OnLine/Secure Security Features User's Guide Version 5.0*, Informix Software, Inc., Menlo Park, CA, April 1993.

[INFO93d] Informix, *Informix-OnLine/Secure Administrator's Guide Version 5.0*, Informix Software, Inc., Menlo Park, CA, April 1993.

[INFO93e] Informix, *DB-Access User Manual Version 5.0*, Informix Software, Inc., Menlo Park, CA, December 1991.

[INFO93f] Informix, *Informix-Star/Secure, Version 5.0 Technical Brief*, Informix Software, Inc., Menlo Park, CA, 1993.

[LUNT88] Lunt, T.F., Schell, R.R., Shockley, W.R., Heckman, Mark, Warren, Dan, *A Near-Term Design for the SeaView Multilevel Database System*, IEEE, 1988.

[LUNT92] Lunt, Teresa, F. "Security in Database Systems: A Research Perspective", *Computers and Security*, Elsevier Science Publishers Ltd., Volume 11, 1992.

[MINA94] Minahan, Tim "Pentagon Aims for 60 percent Commercial Code", *Government Computer News*, Volume 13, Number 1, Page 38, January 10, 1994.

[NCSC89] National Computer Security Center, *DRAFT Trusted Database Management System Interpretation of the Trusted Computer System Evaluation Criteria*, Appendix I-IV, Revised 16 Nov 89.

[NCSC90] National Computer Security Center, *Trusted Product Evaluations: A Guide for Vendors*, NCSC-TG-002, Version-1, June 1990.

[NCSC91a] National Computer Security Center, *Trusted Database Management System Interpretation of the Trusted Computer System Evaluation Criteria*, NCSC-TG-021, Version-1, April 1991.

[NCSC91b] National Computer Security Center, *A Guide to Writing the Security Features User's Guide for Trusted Systems*, NCSC-TG-026, Version-1, September, 1991.

[NCSC92a] National Computer Security Center, *The Design and Evaluation of INFOSEC Systems: The Computer Security Contribution to the Decomposition Discussion*, C Technical Report 32-92, June 1992.

[NCSC92b] National Computer Security Center, *A Guide to Understanding Object Reuse in Trusted Systems*, NCSC-TG-018, Version 1, July 1992.

[NCSC92c] National Computer Security Center, *Guidelines for Writing Trusted Facility Manuals*, NCSC-TG-016, Version 1, October 1992.

[ORAC92a] Oracle Corporation, *Trusted ORACLE Administrator's Guide*, Version 1.0, Oracle Corporation, Redwood City CA, 1992.

[ORAC92b] Oracle Corporation, *ORACLE for HP-UX BLS Installation and User's Guide*, Oracle Version 7.0.9 (Developer's Release), Trusted ORACLE Version 1.0, Oracle Corporation, Redwood City CA, June 5, 1992.

[ORAC92c] Oracle Corporation, *ORACLE RDBMS: Database Administrator's Guide*, Volume 1, Version 7.0, Oracle Corporation, Redwood City, CA 1992.

[ORAC92d] Oracle Corporation, *ORACLE RDBMS: Database Administrator's Guide*, Volume 2, Version 7.0, Oracle Corporation, Redwood City, CA 1992.

[ORAC92e] Oracle Corporation, *ORACLE RDBMS: Database Administrator's Guide*, Volume 3, Version 7.0, Oracle Corporation, Redwood City, CA 1992.

[PFLE89] Pfleeger, Charles, P., *Security in Computing*, PTR Prentice-Hall, Inc., Englewood Cliffs, NJ, 1989.

[SALE93] Salemi, Joe, *Client/Server Computing with ORACLE*, Ziff-Davis Press, Emeryville, CA, 1993.

[SALT75] Saltzer, J.H., Schroeder, M.D., *The Protection of Information in Computer Systems*, IEEE, 1975.

[SCHA84] Schaefer, Marvin and Schell, Roger, "Toward an Understanding of Extensible Architectures for Evaluated Trusted Computer System Products", *Proceedings of the 1984 Symposium on Security and Privacy*, April 1984.

[SCHA91] Schaefer, Marvin, *Reflexions on Current Issues in Trusted DBMS*, Database Security IV: Status and Prospects, North-Holland 1991.

[SHOC87] Shockley, W.R. and Schell, Roger, "TCB Subsetting for Incremental Evaluation", *Proceeding of the Third AIAA Conference on Computer Security*, December 1987.

[STRA93] Straw, Julian, "The Draft Federal Criteria and the ITSEC: Progress Towards Alignment," *Proceedings of the 16th National Computer Security Conference*, Baltimore, MD, 1993.

[T ' NE92] Tanenbaum, Andrew S., *Modern Operating Systems*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1992.

[TROY92] Troy, Eugene, and Ross, Ron, "Perspectives and Progress on International Criteria", *Proceedings of the 15th National Computer Security Conference*, Baltimore, MD, 1992.

[VETT90] Vetter, Linda, Smith, Gordon, and Lunt, Teresa, "TCB Subsets: The Next Step", *Fifth Annual Computer Security Applications Conference*, IEEE, 1990.

[WARE79] Ware, W.H., ed., *Security Controls for Computer Systems: Report of the Defense Science Board Task Force on Computer Security*, AD-A076617/0, Rand Corporation, Santa Monica, CA.,February 1970, reissued October 1979.

INITIAL DISTRIBUTION LIST

Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
Dudley Knox Library Code 052 Naval Postgraduate School Monterey, CA 93943-5002	2
Dr. Ted Lewis Chairman, Code CS Computer Science Department Naval Postgraduate School Monterey, CA 93943	2
Dr. Cynthia Irvine, Code CS/KA Computer Science Department Naval Postgraduate School Monterey, CA 93943	2
Dr. C. Thomas Wu, Code CS/Wq Computer Science Department Naval Postgraduate School Monterey, CA 93943	2
CPT Keith E Muschalek Rt 1 Box 73 Yorktown, TX 78164	1