Common Criteria Security Target for Oracle 7™ Database Server
Release 7.2

March 1998

Common Criteria Security Target for Oracle7™ Database Server
Release 7.2.

March 1998

Author: John Morrissey

Contributors: Duncan Harris, Lydia Frew and Howard Smith

Contents

CHAPTER

1 Introduction

Oracle7 is a relational database management system (RDBMS), providing advanced security functionality for multi-user, distributed database environments. Oracle7 supports client/server application, development, systems management and end-user tools, such as World Wide Web browsers, as well as on-line transaction processing, decision support and data warehousing architectures.

The security functionality in Oracle7 includes granular privileges for enforcement of least privilege, user-configurable roles for privilege management, flexible auditing, stored procedures and triggers for enhanced access control and alert processing and row-level locking.

Evaluated Releases

Table 1 below lists the products and their release numbers that are covered by this security target. The platform configuration is Microsoft Windows NT 3.51 on Intel Platforms.

Evaluated Product Releases	Evaluated Interfaces
Oracle7™ Server & Required Support Files 7.2.2.4	SQL*DBA 7.2.2.4 (Excluding full screen mode)
Distributed Option 7.2.2.4	SQL*Plus™ 3.2.2.0.1
Oracle7™ Utilities 7.2.2.4	OCILIB 7.2.2.4
SQL*Net™ Server/Client 2.2.2.1	
TCP/IP Adapter 2.2.2.1	

Table 1: Evaluated Releases

This Page Intentionally Blank

CHAPTER

2 TOE Description

This section provides a description of the Oracle7 Database Server (Oracle7) and the product features which can contribute to the security of a system.

The security features of the product are explained primarily in [O7_SAG, part IV] and [O7_SC, part VII]. In general, these descriptions correspond to the specifications of IT security functions provided in chapter 6 of this Security Target.

Other security features are described elsewhere in [O7_SC], for example: Part I Chapter 7 Data Integrity; Part IV Chapter 10 Data Concurrency; Part V Chapter 12 Transaction Management; Part VIII Distributed Processing and Distributed Databases; and Part IX Database Backup and Recovery. In general, these features are likely to be relevant to security for many systems, because they contribute to the overall accuracy and integrity of data stored and processed by the system, and the overall availability and reliability of the system.

Structured Query Language (SQL)

As a relational DBMS, the Oracle7 Database Server supports the ANSI/ISO SQL2 standard. All operations performed by Oracle are executed in response to an SQL statement that specifies a valid SQL command.

- Data Definition Language (DDL) statements are statements which create, alter, drop, and rename database objects, grant and revoke privileges and roles, configure audit options; add comments to the data dictionary; and obtain statistical information about the database and its use;
- Data Manipulation Language (DML) statements are statements which modify the data controlled by database objects in one of three ways: by row insertions; by row deletion; or by column update. They include the command to lock a database object.
- Query statements are statements which retrieve data from database objects using the SELECT command.
- Transaction Control statements are statements which manage changes made by

DML statements and help to ensure the integrity of the database. They include commits and rollbacks for individual transactions, and checkpoints for the database;

- Session Control: ALTER SESSION and SET ROLE;
- System Control: ALTER SYSTEM.

Client Server Architecture

Oracle has two major architectural concepts: instance and database. An *instance* consists of a set of DBMS *server* processes, which do the work of the DBMS by executing the Oracle7 Database Server software, and a set of shared memory segments. A *database* consists of a set of files which contain, in addition to some control data, the information which is said to be stored in the database. Each database is an autonomous unit with its own data dictionary that defines the database objects it contains (e.g. tables, valid users and their privileges, etc.). In a distributed system there can be many databases: each database can contain many database objects, but every database object is stored within only one database.

An instance is therefore an active entity, and a database is passive. In order for users to access the database, the instance must be started and must *mount* and *open* the database for use. A database is *persistent*: it has an indefinite lifetime from the time it is created, and the database files and contents exist independently of whether the database is mounted to an instance, and whether the underlying platform is running. The lifetime of an instance can be indefinite, from when it is started to when it is shut down, and is dependent on whether the underlying platform is running.

Each database user needing to access the database (more precisely, the *client* application program acting on the user's behalf) then *connects* to the instance (and hence also to the database mounted by the instance). If the user is defined as a valid user and has the access rights to create a session in the database, then the instance will create a *database session* for the user. While connected, the user can make requests to the instance to read and write information in the database. The instance services each request, performing the read and write accesses to database objects and returning data and results to the user, in accordance with the user's access rights to database objects and resources (as enforced by the security features described below) and other constraints configured by the database administrator.

A user connects *directly* to a *local* instance. Subsequently a user may connect *indirectly* to a *remote* instance, through the local instance to which he remains connected directly. The database mounted by a local instance is said to be a *local* database; that mounted by a remote instance is said to be a *remote* database. The terms *local* and *remote* are relative to the user's connection: one user may be connected to a database via a remote connection while simultaneously another user may be connected to the same database via a local connection.

In order to provide users with access to a remote database, *database link(s)* must be created and configured in the users' local database(s) such that, when a user connected to his local database attempts to read data in a remote database, Oracle validates that the user is defined with the same name in the remote database. Each instance is autonomous: when a user connects to a remote database the remote instance enforces security based on the access rights of the user as defined in that database.

Identification and Authentication

Oracle7 always identifies authorised users of a particular Oracle7 database. In the evaluated configuration authentication of an identified relies on the previously performed identification and authentication of the underlying operating system or network services in place of authentication by Oracle7.

Specially authorised users may connect to the database to undertake database administrator (DBA) functions such as starting up or shutting down an instance. These users must be in possession of platform specific DBA access rights; for example on a UNIX platform an operating system user must be a member of a special UNIX group, normally the DBA group. When an operating system user with such platform specific DBA access rights wants to undertake privileged DBA operations, he connects to the database through a special keyword: INTERNAL, AS SYSDBA or AS SYSOPER.

Access Control

Oracle7 implements Discretionary Access Control (DAC) security by default. DAC allows authorised users to selectively share information with other users. Oracle's DAC can be used to enforce need-to-know style confidentiality as well as control data disclosure, entry, modification, and destruction.

Discretionary access control mediates access by users to *database objects* and data contained within those objects. Examples of such objects provided by Oracle7 are *tables*, *views*, *stored procedures*, and *triggers*. Oracle7 performs this DAC mediation through privileges: in order gain access to certain information the user must be the owner of the information or have the appropriate *database privilege*. A database privilege is either an *object privilege*, which is permission to access an object in a prescribed manner (for example, for read and update only), or a *system privilege*, which is permission to perform a specific operation in the database (for example, to create a table). Because database privileges are granted to users at the discretion of other users, this type of security is termed *discretionary*. Oracle7 ensures that users who attempt to gain access to objects or to perform particular operations in a database have the appropriate database privileges.

User Management

Suitably privileged Oracle users can create uniquely identified users of a database. They can also grant or revoke database privileges and roles (groups of privileges) to or from a user or role. Similarly, suitably privileged users can drop (delete) other users and all the objects they own.

If it is necessary to prevent one or more users from connecting to a database, but it is not desirable to delete their user accounts, then appropriately privileged Oracle users can prevent a user from subsequently connecting to a database by revoking the privilege required to create a DBMS session. The privilege to connect to the database can be restored when and if required by suitably authorised users.

Program Units

Program Units is the collective name for stored procedures, functions, packages, and triggers.

Stored Procedures and Functions

A *stored procedure* is a user-defined set of PL/SQL and SQL statements that are grouped together as a unit to perform a specific function. These statements are stored in Oracle in source and executable forms.

Users with the appropriate privileges can execute a particular stored procedure or function. When a user executes a stored procedure, the stored procedure operates with the discretionary privileges (though not the identity) of the procedure owner and not the executor. Thus system developers can enforce granular discretionary security by restricting users to accessing data or specified subsets or aggregations of that data only

through pre-defined stored procedures and functions.

For example, a suitably authorised user can grant access to a procedure that updates a table, but not need to grant update access to the table itself. Other users granted the execute privilege on the procedure can run the procedure, but unless they have additional privileges they cannot arbitrarily manipulate data in the table in any other way. This can also be used to limit, for example, the time of day and day of week (perhaps only regular business hours) at which, or the terminal location from which the specified updates can be made through the stored procedure.

Stored functions are identical to stored procedures except that functions return a value.

Packages

A *package* is a user defined set of stored procedures, functions, and variable declarations that are stored as a unit.

When users are granted the EXECUTE object privilege on a package, they are implicitly granted the EXECUTE object privilege on all the stored procedures and functions within the package. Thus packages, like roles, are helpful in simplifying security administration by providing a mechanism for grouping together functions and procedures.

Triggers

Oracle allows appropriately privileged users to define database procedures called *triggers* that are implicitly executed when data is inserted, updated, or deleted from a table. When a user modifies data in a table in a way which causes a database trigger to fire, the trigger operates with the DAC privileges of the owner of the table and trigger and not the executor. Thus system developers can enforce granular discretionary security by restricting users to accessing data only through a database trigger. However, it should be noted that such code may be security enforcing in the context of the host application and may therefore need to be subjected to its own independent evaluation.

For example, a system developer can write a trigger to record to another table a history of the specific values that are changed by users modifying the data in the table. While the owner of the table and the trigger must have access to the history table, the user whose update of the table fires the trigger does not need to be granted access to the history table; thus the user cannot maliciously or unintentionally manipulate data in the history table. (See the section *Enhanced Database Auditing*.)

Pipes and Alerters

Pipes allow different DBMS sessions (users) in the same database to communicate with each other. *Alerters*, which make use of pipes, provide for the asynchronous notification of database events. Through the use of Alerters, an application can cause itself to be notified whenever values of interest in the database are changed. For example, if the number of on-hand replacements for a mission-critical item dips below a threshold level, an appropriate user can be notified immediately.

Other Database Objects

Indexes and *Clusters* are database objects which can be created to increase the performance of data retrieval. *Sequences* are another database object which generate serial lists of unique numbers for columns of a database's tables. Sequences simplify application programming by automatically generating unique numerical values for the rows of a single table or multiple tables. The creation, modification, use, and deletion of all three of these types of objects can be controlled through discretionary access control privileges in sensible ways to ensure an organisation’s security policy can be fully implemented.

Accounting and Auditing

Oracle7 also provides a number of features and functions to enable accountability of actions taken by users of the database. Oracle7 does this by providing accounting and auditing features which are designed to be as granular and flexible as possible to ensure that exactly what needs to be accounted and audited, as dictated by the application or system security policy, is recorded, but nothing more. This helps to ensure that the size of audit trails remain manageable and the important records easily accessible. Oracle7 also provides capabilities to permit accounting and auditing plans to be quickly enabled to implement crisis plans.

By default, Oracle7 records no accounting or auditing information, except for a few privileged operations by administrators. However, Oracle7 can be configured to write accounting and auditing information to its own database audit trail or to the underlying operating system's audit trail (or to a specified operating system file if no official operating system audit trail exists). If configured to write information to its own database audit trail, the powerful SQL data manipulation facilities of the DBMS can be used by appropriately privileged users to perform selective accounting and audit analysis quickly and efficiently. Alternatively, if configured to write to the audit trail (or specified file) in the underlying operating system, platform services may be used to consolidate and analyse the audit trail from the database with audit trails from other system components to provide a comprehensive accounting and auditing portrait for the system as a whole.

In a system with two or more physical databases, whether standalone or distributed, Oracle undertakes accounting and auditing of actions performed in each database in accordance with the accounting and auditing instructions specified in that database.

Accountability Features

Oracle ensures that relevant information about operations performed by most users can be recorded so that the consequences of those operations can later be linked to the user in question, and the user held accountable for his actions. No information about actions performed by users connected as the special user SYS or through the special keywords INTERNAL, AS SYSDBA and AS SYSOPER are recorded, except for their attempts to make these special connections, to startup an instance and to shutdown an instance.

Oracle ensures that sufficient information can be recorded about both routine and exceptional events so that investigations can determine if security violations have actually occurred, and if so what data or other database resources were compromised. Specifically Oracle ensures that appropriately privileged users can:

- Selectively account for the actions of one or more users;
- Selectively account for actions by one, several, or all users making use of one or more system privileges;
- Selectively account for successful, unsuccessful, or both successful and unsuccessful attempts to exercise access rights to one or more Oracle objects;
- Selectively account for successful, unsuccessful, or both successful and unsuccessful attempts to perform certain types of operations by one, several or all users;
- Selectively account for the creation or destruction of one or more Oracle objects; and
- Selectively account for actions by authorised users affecting the security of Ora-

cle.

Auditing Features

Oracle ensures such accountability by writing information to an audit trail in the database or the underlying platform (*standard auditing*) and also by providing the mechanism of PL/SQL database triggers which can be used to record specific changes to data values and other specialised audit data to database tables (*enhanced database auditing*).

The special database user SYS owns the database audit trail table, and hence users connected as SYS (or through the keywords INTERNAL, AS SYSDBA and AS SYSOPER) may read and write all rows in the audit trail table. Users connected in this way may perform database audit trail analysis and clear out old audit trail records (rows). Any normal user granted appropriate object privileges on the database audit trail, or appropriate system privileges, may also access the database audit trail to perform audit trail analysis and clear out old audit records, but such accesses may themselves be audited.

Standard Auditing

Using Oracle standard auditing, appropriately privileged users can request auditing of any number of actions in each of three categories:

- *By Statement*
  Auditing specific types of SQL statements including database connections and disconnections. Statement auditing can be set to audit one, several, or all users.
- *By Object*
  Auditing specific statements on database objects for all users.
- *By Privilege*
  Auditing use of specific system privileges. Privilege auditing can be set to audit one, several, or all users.

Appropriately privileged users can further focus each auditing request by specifying auditing for only successful, only unsuccessful, or both successful and unsuccessful attempts. Such users can also specify for most audit options that audit records be created *by session* or *by access*: by session results in only a single record for an audited action for the duration of a DBMS session (the time between when a user connects to and disconnects from a Oracle database); by access results in a record for every occurrence of an audited action.

Oracle also permits authorised users to assign default object auditing options which will automatically be used for any new objects which are created.

Oracle's standard auditing permits accountability information to be written to the database audit trail or to the audit trail of the underlying operating system or network services. The Oracle audit trail always includes the following elements when they are meaningful for the audited event:

- User;
- Session Identifier;
- Terminal Identifier;
- Name of Object Accessed;
- Operation Performed;
- Completion Code of Operation;

- Date and Timestamp;
- System Privilege Used.

If Oracle writes to the database audit trail, then the powerful SQL data manipulation facilities of the DBMS can be used by appropriately privileged users to perform selective audit analysis of relevant database operations, user actions, uses of privilege, and object accesses in a secure manner. Oracle provides a number of pre-defined queries (views) on the database audit trail to assist in such audit analysis. Nonetheless, the database audit trail is protected against unauthorised query, entry, modification, and destruction.

If Oracle is configured to write to an operating system or network services audit trail, then platform services can be used to consolidate and analyse the database audit trail with audit trails from other system components to provide a comprehensive auditing portrait for the system. Alternatively, the audit data in the operating system or network services audit trail could be loaded securely into a Oracle database for comprehensive audit analysis using the powerful SQL data manipulation facilities of the DBMS.

Enhanced Database Auditing

Using PL/SQL database triggers (and, optionally, PL/SQL stored procedures or functions), appropriately privileged users can audit specific changes made to the values of data in database tables. *It should be noted however that any such code may be security enforcing in the context of the host application and therefore may need its own independent evaluation.*

Such audit records can be generated for each *row* successfully inserted into, updated in, or deleted from a specified table. These audit records can capture the actual changes to the data made. Similar audit records can be generated for each *statement* which successfully inserts, updates, or deletes rows in a specified table.

Database triggers can be defined to audit based on content or context of the change to data:

- *Content*
  For example, only updates to the bonus column of an employee table could be audited.
- *Context*
  For example, only modifications of data occurring outside of Monday to Friday, 0900 to 1700.

Oracle's PL/SQL database triggers permit accountability information to be written securely to the database audit trail or to other specified database tables. The accountability information captured through database triggers may include:

- Old and new values of changed data;
- User;
- Session Identifier;
- Terminal Identifier;
- Name of Object Accessed;
- Operation Performed;
- Date and Timestamp.

The SQL data manipulation facilities of the DBMS can be used by appropriately priv-

ileged users to perform selective audit analysis of the accountability information resulting from enhanced database auditing in a secure manner. The audit trail resulting from Enhanced Database Auditing can be protected, like all database tables, against unauthorised query, entry, modification, and destruction.

Accuracy and Integrity

The Oracle7 Database Server provides mechanisms to ensure that the consistency and integrity of data held in a database can be maintained. These mechanisms are transactions, concurrency controls, and integrity constraints. Transactions ensure that updates to the database occur in well-defined steps that move the database from one consistent state to another. Transactions and concurrency controls together ensure that multiple users can have shared access to the database with consistent and predictable results: each user sees a consistent state of the database and can make updates without interfering with other users. Integrity constraints ensure that the values of individual data items are of the defined type and within defined limits, and that defined relationships between database objects are properly maintained.

Availability and Reliability

The Oracle7 Database Server provides mechanisms for ensuring that databases are available to authorised users when needed, that services are provided in a reliable manner, and that the competing needs of multiple users for database resources can be effectively managed. These mechanisms are backup, recovery, tablespace quotas, and resource profiles. Backups prepare for recovery by taking full or partial copies of database data. Recovery restores the database to the most recent consistent state following a system failure or other contingency. Individual users can be prevented from excessively consuming resources, by means of:

- tablespace quotas; *and*
- resource profiles.

Security Administration and Management

Oracle enables the separation of functions or duties required by sophisticated system security policies to be implemented correctly by providing a granular division of what are sometimes known as *database administrator*, *system security officer*, and *database operator* duties into over eighty distinct, separately managed Oracle privileges.

Oracle facilitates correct privilege administration by enabling privileges to be grouped together into database roles. The benefits of Oracle database roles include:

- Reduced privilege administration;
- Dynamic Privilege Management;
- Least Privilege enforcement;
- Database application privilege management;
- Consistent system security policy;
- Secure Data Exchange;

- Secure Distributed Processing and Distributed Databases.

Reduced privilege administration

For example, rather than explicitly granting the same set of privileges to several users, the privileges for a group of related users can be granted to a role, and then only the role needs to be granted to each member of the group. Roles permit numerous Oracle privileges to be granted or revoked with a single SQL statement.

Dynamic privilege management

If the privileges of a group of users must change, only the privileges of the role(s) need to be modified instead of the privileges granted to every user. The security domains of all users granted the group's role automatically reflect the changes made to the role.

Least privilege enforcement

The roles granted to a user can be selectively enabled (available for use) or disabled (not available for use). This ensures control of a user's privileges in any given situation and helps to prevent accidental or malicious use of those privileges which could result in unintended disclosure, entry, modification, or destruction of data. In addition, roles can be marked to be enabled when a specific user creates a DBMS session (by default) or not, further providing least privilege enforcement.

Database application privilege management

Because the Oracle data dictionary records which roles have been granted to, and are enabled for, the current user, database applications can be designed to query the dictionary and automatically enable and disable selective roles when a user attempts to execute applications.

Consistent system security policy

To enable centralised implementation of privilege management in a system of which Oracle may be only one component, Oracle also provides for linking database roles to platform-specific group access controls. In this way, database roles can only be enabled by users if they are a current member of the appropriate group in the underlying platform. This helps to ensure a correct and consistent implementation of a system-wide security policy.

Secure Data Exchange

In a secure, distributed database environment it is important to ensure that data can be moved out of one database and re-inserted into the same or a different database whilst maintaining the data integrity.

Oracle enables secure exporting of information from a database into an operating system file. Only appropriately privileged users may export information to which they do not normally have read access. Similarly, Oracle enables secure importing of information into a database from Oracle-generated operating system export files. Only appropriately privileged users may import information into database tables to which they do not normally have write access.

Secure Distributed Processing and Distributed Databases

The features provided for distributed processing and distributed databases are summarised in [O7_DSV1, section 1]. The security implications are explained in [O7_DSV1, section 6] Database Security in a Distributed System. The features include the following:

- Direct and indirect connections, as described in [O7_DSV1, p.1-19];

- Remote and Distributed Queries and Transactions;
- Transaction Recovery Management.

Distributed processing involves some combination of client-server and server-server interaction.

Oracle is based on a client-server architecture, in which the database application and the database are separated into two parts: a front-end or client part, and a back-end or server part. The client executes the database application that accesses database information and interacts with the user. The server executes the Oracle Database Server software and provides the functions required for concurrent shared access to an Oracle database.

Oracle can be configured so that both the client and server parts are executed on the same computer hardware component. Distributed processing is the use of more than one hardware component, so that the client part and the server part run on separate, dedicated hardware components.

Server-server interaction supports distributed databases. In a distributed environment there can be multiple databases. A given user can access at least one database in the distributed system, and possibly several. The database that a user is directly connected to is termed the user's local database. A database object contained in one database may contain a reference (called a database link) to an object contained in another database. A database link is effectively a pointer from one database to another: user access to the original object is automatically redirected to the remote object transparently to the user. Any databases that the user accesses indirectly in this way are termed remote databases.

The Oracle7 Server provides site autonomy [O7_DSV1, p.1-19]. This means that each server participating in a distributed system is administered independently (for security and backup operations) from the other servers. Access to each database is managed by an instance of the Oracle7 Database Server: there is a one-to-one relationship between databases and instances. Each instance is autonomous and enforces the security policy in its entirety for the database and all objects contained in the database. Therefore, in relation to the threats and security objectives stated below, there are no security features provided by the Oracle7 Database Server which are specific to secure distributed processing; for example, database links are not considered to be security features.

In order to maintain its autonomy in client-server interaction and server-server interaction, each server receiving a request must be able to obtain the security attributes of the requesting client process (the subject, in security terms) making the request.

CHAPTER

3 Security Environment

Threats

As per [CDBMS PP, 3.1].

Organisational Security Policies

As per [CDBMS PP, 3.2];

Secure Usage Assumptions

Connectivity Assumptions

As per [CDBMS PP, 3.3.1] with the addition of:

A.APPLICATIONS All executables which may be activated along side the TOE are assumed to maintain the security policy constraints, in particular:

a) no applications shall be allowed access to the network; *and*

b) all applications making use of Oracle API, shall be "well behaved" with respect to that interface (i.e. shall use it in the manner proscribed in the user documentation).

Physical Assumptions

As per [CDBMS PP, 3.3.2].

Personnel Assumptions

As per [CDBMS PP, 3.3.3].

This Page Intentionally Blank

CHAPTER

4 Security Objectives

IT Security Objectives

As per [CDBMS PP, 4.1].

Non-IT Security Objectives

As per [CDBMS PP, 4.2] with the following additional objective.

O.CONFIGURE The administrators of the database must ensure that the TOE is configured as follows:

a) The OS_AUTHENT_PREFIX initialisation parameter must be the same on each node.

b) Each database must be given a database name which is unique within the node on which the database resides.

c) After creating and setting up a database, all database user accounts must be configured to use DBMS Identification and OS Authentication. This includes any pre-defined accounts (such as SYS and SYSTEM) and any demonstration accounts (such as SCOTT) created during installation. This is the only reconfiguration of the SYS account allowed in the evaluated configuration.

d) The SQL92_SECURITY initialization parameter must be set to TRUE.

e) The only database links included in the evaluated configuration are anonymous database links.

f) Only those system users who are authorised to gain access to the DBA, OPER or INTERNAL privileged database accounts must be allowed to gain access to the correspondingly identified accounts in the underlying operating system.

This Page Intentionally Blank

CHAPTER

5 IT Security Requirements

TOE IT Security Functional Requirements

The following table lists the functional components included in this ST.

Component	Name
FIA_UID.1	Timing of Identification (*refined*)
FIA_ATD.1	Unique User Attribute Definition
FIA_USB.1	User-Subject Binding
FDP_ACC.1	Subset Object Access Control
FDP_ACF.1	Single Security Attribute Access Control
FDP_RIP.1	Subset Residual Information Protection on Allocation
FMT_MSA.1	Basic User Attribute Administration
FMT_MSA.3	Static Attribute Initialisation
FMT_MTD.1	Management of TSF data
FMT_SMR,1	Security Management Roles
FMT_REV.1	Basic Revocation
FRU_RSA.1	Maximum Quotas
FTA_MCS.1	Basic Limitation on Multiple Concurrent Sessions

Table 2: List of Security Functional Components

Component	Name
FAU_GEN.1	Audit Data Generation
FAU_GEN.2	User Identity Generation
FAU_SAR.1	Audit Review
FAU_SAR.3	Selectable Audit Review
FAU_SEL.1	Selective Audit
FAU_STG.1	Permanent Audit Trail Storage

Table 2: List of Security Functional Components

Each of the above components is as specified in [CDBMS PP], with operations being completed for the following SFRs:

- FIA_UID.1.1;
- FIA_ATD.1.1;
- FIA_RIP.1.1;
- FMT_SMR.1.1;
- FRU_RSA.1.1;
- FAU_GEN.1.1;
- FAU_SAR.3.1;
- FAU_SEL.2.1.

Refer to [CDBMS PP, 5.1] for a full specification of the SFRs, and to section A.2 below for details of how the above operations have been completed in this ST.

IT Assurance Requirements

The target assurance level is EAL4 as defined in Part 3 of the CC. No augmented assurance requirements are defined.

Security Requirements for the IT Environment

As per [CDBMS PP, 5.3], the requirement for the underlying operating system target assurance level is increased to EAL4.

Minimum Strength of Function

As per [CDBMS PP, 5.4].

CHAPTER

6 TOE Summary Specification

TOE IT Security Functionality

This section contains the definitions of the TOE IT security functions which satisfy the SFRs.

Identification and Authentication

F.IA.PRE Oracle shall only allow users to:

a) obtain the current Oracle version string and version number;

b) establish a connection;

c) receive error messages upon error.

before identifying the user.

F.IA.UID Each database user is uniquely identified for a particular database.

F.IA.CNF The TOE will allow only a suitably authorised user to create a database user, and/or set and alter the default security attributes of a database user.

F.IA.IDE For each interaction between a user and the TOE following the successful creation of a database session, the TOE is able to establish the identity of the user. A subject can only submit requests to a Server and receive responses (information) from a Server while the subject is establishing or connected to an instance during the course of a database session.

F.IA.CSA The TOE will create a database session as a DBA user or OPER user only if the requesting subject has the platform-specific

access rights for OSDBA and OSOPER, respectively, as defined in [O7_SAG] Chapter 1 OSOPER and OSDBA.

F.IA.CON The TOE will create a database session as a normal user only if the requesting subject's user identifier, prefixed by the value of the OS_AUTHENT_PREFIX initialisation parameter, matches a database user identifier.

F.IA.CSN The TOE will create a database session as a normal user only if the CREATE SESSION privilege is held by the database user.

F.IA.ATT The data dictionary contains a unique set of security attributes for each user including their privileges, roles and resource limits that can be displayed and modified using standard SQL commands.

Access Control

Database Resources

F.LIM.CNF The TOE will allow only a suitably authorised user to:

a) alter the default Resource Profile for a database;

b) create and alter specific Resource Profiles and assign and reassign them to each individual database users.

F.LIM.POL When a user attempts to use a database resource that is subject to controls specified by Resource Profiles, the TOE will enforce the limits specified by the resource profile (if any) explicitly assigned to the user, otherwise it enforces the limits specified by the default Resource Profile for the database.

F.LIM.NSESS The TOE prevents a user from creating more than the maximum number of concurrent sessions specified for that user for an instance of the TOE.

The maximum number of concurrent sessions for a specific user is configurable and can be different for each user. Resource limits are not enabled by default but once enabled, this limit can be set to any value, such as 1 (a single session) by the administrator.

F.LIM.TIME If a user exceeds the specified CONNECT_TIME or IDLE_TIME resource limits by the (OS specific) amount for a single session then the TOE will terminate the session when the user attempts an operation.

F.LIM.RSESS If a user attempts to perform an operation that exceeds the specified resource limits for a single session then the TOE will:

a) terminate the operation;

b) force the termination of the session.

F.LIM.RCALL If a user attempts to perform an operation that exceeds the specified resource limits for a single SQL statement then the TOE will terminate the operation.

Object Access Control

F.OAC.OBID The TOE ensures that every object created in a database is uniquely identified in that database. Specifically, each schema

object owned by a normal user is uniquely identified within the user's schema[1].

F.OAC.OBREF The TOE correctly resolves every reference to a database object that conforms to the Object naming rules specified in [O7_SQLRM, chapter 2], including references via database links[2].

F.OAC.SUA For normal users, the TOE enforces DAC on database objects based on the following subject attributes:

a) the identity of the user associated with the database session;

b) the system privileges and object privileges which are effective for the database session.

F.OAC.OBA For normal users, the TOE enforces DAC on database objects based on the following object attributes:

a) the identity of the owner of the object;

b) the object privileges which have been granted on the object.

F.OAC.POL The TOE enforces the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

a) If the user is the owner of the object then the requested access is allowed.

b) If the database session has the necessary object privileges effective for the object then the requested access is allowed. The object privileges relevant to different types of objects are specified in [O7_SAG] Chapter 13 Object Privileges, and provide the ability to restrict a user's access to an object to those operations which do not modify the object.

c) If the database session has the necessary system privileges effective then the requested access is allowed. The system privileges relevant to different types of database-wide and schema-specific operations are specified in [O7_SAG] Chapter 13 System Privileges, and provide the ability to restrict a user's use of operations to those operations which do not modify objects.

---

1. The owner of an object is the owner of the schema containing the object, not necessarily the user who created the object. More precisely, unique identification is by object type as well as object name within a schema.

2. A reference to a database link (e.g. CONNECT /@otherdb or SELECT * FROM TBL@otherdb) will be correctly resolved to the referenced database. A database object can be uniquely identified in a distributed system, because it is uniquely identified in the database, and the database is unique in the system. The threat is that failure to uniquely identify objects and user accounts could result in reading, creating, modifying or destroying the wrong object (or copy of an object) if the user has the same access rights in each database.

d) If the user is the DBA user (the database session has the privilege to override the access controls) then the requested access is allowed.

e) If the user is the OPER user and the operation is one of those specified in [O7_SAG] Chapter 1 OSOPER and OSDBA, for the OSOPER role then the requested access is allowed.

F.OAC.OR Upon allocation of a resource to schema and non-schema objects, any previous information is unavailable. In Oracle, there is no way to access an object once it has been deleted, i.e. the resources have been returned to the TOE. This is because any references to it no longer exist and, even if they were recreated, they would never be associated with the previous, non-existent object.

All objects have a unique ID. Even if a deleted object is recreated using the same name, the object ID would be different.

Schema and non-schema objects are defined in [O7_SQLRM, chapter 2].

Privileges and Roles

Granting and Revoking Privileges and Roles

F.APR.GOP A normal user (the grantor) can grant an object privilege to another user, role or PUBLIC (the grantee) only if:

a) the grantor is the owner of the object; *or*

b) the grantor has been granted that object privilege with the GRANT OPTION.

F.APR.ROP A normal user (the revoker) can revoke an object privilege from another user, role or PUBLIC (the revokee), and any further propagation of that object privilege started by the revokee, only if the revoker is the original grantor of the object privilege.

F.APR.GRSP A user (the grantor) can grant a system privilege to another user, role or PUBLIC (the grantee), and revoke a system privilege from the grantee, only if:

a) the grantor (or revoker) is the DBA user; *or*

b) the database session of the grantor (or revoker) has the GRANT ANY PRIVILEGE privilege effective; *or*

c) the grantor (or revoker) has been granted that system privilege directly with the ADMIN OPTION.

F.APR.GRR A user (the grantor) can grant a role to another user, role or PUBLIC (the grantee), and revoke a role from the grantee, only if:

a) the grantor is the DBA user; *or*

b) the database session of the grantor (or revoker) has the GRANT ANY ROLE privilege effective; *or*

c) the grantor (or revoker) has been granted that role with the ADMIN OPTION[3].

Enabling and Disabling Roles	**F.APR.DER**	A role can be granted to a user in one of the following ways: a) As a non-default role, in which case the user must explicitly enable the role during a database session in order for any other roles within that role to be enabled and any privileges within that role to become effective for that user. b) As a default role, in which case the role will be enabled automatically for each database session created by that user.
	F.APR.EDR	During a database session the user can control which roles are effective at any time during the course of the database session by enabling and disabling the roles which have been granted to that user (where the role may have been granted directly to the user or granted indirectly to the user through other roles[4]), subject to the following restrictions which apply to implicit remote sessions: a) The non-default roles granted to a user in a remote database cannot be enabled while the user is connected to the remote database. b) The default roles granted to a user in a remote database cannot be disabled while the user is connected to the remote database.
Gaining and Losing Privileges	**F.PRI.NULOSE**	The TOE will cause a normal user to lose a privilege if: a) the privilege was granted to the user directly and is revoked from the user; *or* b) the privilege was granted indirectly via the PUBLIC user group and is revoked from PUBLIC; *or* c) the privilege was granted to the user indirectly via one or more roles, and is revoked from those roles.

3. This includes the case where the grantor is the user who created the role - see [OR_SAG] p.12-16: "When a user creates a role, the role is automatically granted to the creator with the ADMIN OPTION." and the warning on p.12-12 (Set Default Roles) which adds the fact that a role is automatically granted to its creator as a default role.

4. When a role that contains other roles is enabled all the indirectly granted roles are implicitly enabled.

F.PRI.RPLOSE A privilege will be taken away from a specified role or the PUBLIC user group if:

a) the privilege was granted to the role or PUBLIC directly and is revoked from the role or PUBLIC; *or*

b) the privilege was granted to the role or PUBLIC indirectly via one or more other roles, and is revoked from those roles.

F.PRI.SINIT When a user session is created a privilege will be effective only if:

a) the privilege has been given, and has not been taken away from, the user or PUBLIC at the time the session is created; *or*

b) the user has been granted directly or indirectly at least one role which holds the privilege, and that role is included in the set of default roles for the user; *or*

c) the privilege is granted to a role that is enabled for the user session.

F.PRI.SLOSE A privilege will cease to be effective for a normal user session if:

a) the privilege was directly granted to the user and the privilege is directly revoked from the user; *or*

b) the user holds the privilege by virtue of being granted a role directly or indirectly, and the user disables that role; *or*

c) the user holds the privilege by virtue of being granted a role directly or indirectly, and the privilege is revoked from that role.

F.PRI.XVP A suitably authorised user can provide other users with access to proxy mechanisms (namely Views and Program Units) which will act on behalf of the owning user (by executing with the privileges of the owning user) to allow other users to have controlled access to specified aggregations of data.

Audit and Accountability

F.AUD.SOM When standard auditing is enabled (as DBMS or OS Auditing) for an instance, the TOE will:

a) write an audit record for every occurrence of an auditable event other than CONNECT and DISCONNECT; and

b) write an audit record for every pair of CONNECT/ DISCONNECT events.

F.AUD.SEV The TOE will allow a suitably authorised user to specify which events for a database are auditable, as follows:

a) by use of DDL statements, for all users or for specified users;

b) by use of DML statements;

i. for specified Object Privilege Objects;

ii. for all Object Privilege Objects subsequently created, by default;

c) by use of system privileges, for all users or for specified users;

d) for each event of type b) by session or by access, i.e. only one audit record written for each auditable event that occurs in the same session or one audit record written for each auditable event. For events of type c) by session or by access, unless a DDL statement when always by access;

e) for each event of type a), b) and c) by outcome, i.e. success, failure, or both.

F.AUD.ALW If an operating system audit trail is provided and is enabled, and irrespective of the TOE's audit configuration, the TOE will audit every successful occurrence of the following events to the operating system audit trail:

a) start-up;

b) shut-down;

c) successful connection through the keywords INTERNAL, AS SYSDBA or AS SYSOPER.

F.AUD.CNF The TOE will allow only a suitably authorised[5] user to set or alter the audit configuration for a database.

F.AUD.ACC The TOE will allow suitably authorised users to select by criteria audit information from the database audit trail, as follows:

a) the normal user SYS and the DBA user can view all audit records;

b) the owner of an object can view the audit records relating to that object.

F.AUD.DEL The TOE will allow only a suitably authorised[6] user to delete audit records from the Database Audit Trail.

F.AUD.INF The TOE will record the following information into each Database Audit Trail record, provided that the information is meaningful to the particular audited event:

Date and time of event; username; instance ID for the Oracle instance where the user is accessing the database; session identifier; terminal identifier of the user’s terminal; name of

5. By default, the TOE allows only the DBA user to set and alter the audit configuration. It is possible for the DBA user to grant the relevant privileges to other users, but it is assumed that the DBA user will not do this in practice.

6. By default, the TOE allows only the DBA user to delete rows from SYS.AUD$. It is possible for the DBA user to grant the relevant privileges to other users, but it is assumed that the DBA user will not do this in practice.

object accessed; operation performed or attempted; outcome of the operation; system privileges used.

In particular:

a) when a user attempts a connections to a database, whether successful or not, at least the following information is recorded when the TOE is configured to audit connection attempts: date and time of event, username, instance ID for the Oracle instance where the user is accessing the database, session identifier, terminal identifier of the user's terminal, outcome of the connection attempt;

b) when a user attempts to access any database object, whether successful or not, at least the following information is recorded when the TOE is configured to audit such access attempts: date and time of event, username, name of object accessed, operation performed or attempted, outcome of the operation;

c) when a user attempts to create or drop any database object, whether successful or not, at least the following information is recorded when the TOE is configured to audit such create or drop actions: date and time of event, username, name of object to be created or dropped, operation performed or attempted, outcome of the operation;

d) when a user attempts to affect the security of the TOE, by, for example, startup up and shutting down an instance of the TOE, creating new, modifying existing or dropping old user accounts, tablespaces, databases, rollback segments, etc. as the TOE permits at least the following information is recorded when the TOE is configured to audit such actions: date and time of event, username, name of object accessed, operation performed or attempted, outcome of the operation.

F.AUD.VIEW Oracle provides both the SQL language and built-in views, based on the underlying audit trail table SYS.AUD$, with the ability to both view and search the audit data.

Security Mechanisms and Techniques

No specific security mechanisms or techniques are claimed or required for this TOE.

Assurance Measures

The target assurance level is EAL4, which exceeds the assurance requirement of EAL3 as stated in [CDBMS PP]. No specific assurance measures are claimed.

ANNEX

A Protection Profile Claims

The TOE conforms to the Commercial Database Management System Protection Profile [CDBMS PP].

PP Refinements

Refinements have been made to the following Security Functional Requirements. All of these refinements, which are italicised, are in accordance with the assignment statements in [CDBMS PP].

Identification and Authentication

FIA_UID.1.1 The TSF shall allow users to:

a) *obtain the current Oracle version string and version number*

b) *establish a connection*

c) *receive error messages upon error*

before identifying the user.

FIA_ATD.1.1 The TSF shall maintain a set of privileges, roles and resource limits belonging to individual users.

User Data Protection

FDP_RIP.1.1 The TSF shall ensure that upon the allocation of a resource to *schema and non-schema objects* any previous information content is unavailable.

Security Management

FMT_SMR.1.1 The TSF shall maintain the following roles:

a) *SYSDBA; and*

b) *SYSOPER.*

Resource Utilisation

FRU_RSA.1.1 The TSF shall enforce quotas limiting the maximum quantity of:

a) *CPU time for a session*

b) *CPU time for a call*

c) *total elapsed time of a session (connect time)*

d) *periods of continuous inactivity during a session (idle time)*

e) *the number of data blocks read in a session (logical reads)*

f) *the number of data blocks read per call*

g) *the total resource cost for a session*

that an individual user can use during a database session.

Security Audit

FAU_GEN.1.1 The TSF shall be able to generate an audit record of the following auditable events:

a) Start-up and shutdown of the audit functions;

b) All auditable events for the basic level of audit as defined in all functional components included in the PP/ST (see Table 2 in [CDBMS PP]); and

c) Based on all functional components included in the PP/ST,

i. *Every pair of `CONNECT/DISCONNECT` events*

ii. *DDL statements*

iii. *DML statements*

iv. *Use of system privileges*

v. *Database startup and shutdown (to OS audit trail only)*

vi. *Successful connection as `INTERNAL`, `AS SYSOPER` and `AS SYSDBA` (to OS audit trail only).*

FAU_SAR.3.1 The TSF shall provide audit review tools with the ability to perform searches and sorting of audit data based on *all database fields, including the ROWNUM pseudocolumn with sorting as defined in [O7_SQLRM, 4-405, ORDER BY Clause]*.

FAU_SEL.1.1 The TSF shall be able to include or exclude auditable events from the set of audited events based on the following attributes:

a) Event Type;

b) *Use of system privilege*;

c) *By outcome, i.e. success, failure or both.*

PP Additions

There are no additional threats, organisational security policies or secure usage assumptions included in this ST.

There are no additional IT security objectives included in this ST.

The non-IT Security objective O.CONFIGURE has been added to supplement those objectives already defined in [CDBMS PP]. This objective has been included to cover the configuration issues that are specific to Oracle, and does not conflict with any other non-IT security objective.

There are no additional functional components included in this ST, either on the TOE or on the IT environment.

The assurance requirement specified in this ST is EAL4. This includes all assurance requirements in [CDBMS PP] (which mandates EAL3).

Rationale for PP Claim

Section 5.1 above lists all SFRs included in this ST; this list includes all SFRs specified in the CDBMS PP. Section A.2 above demonstrates how the uncompleted PP operations are fulfilled in this ST.

Table 3 below provides a mapping of the TOE IT security functions to SFRs, showing how the SFRs are satisfied.

SFR	IT Security Functions	Notes
FIA_UID.1.1	F.IA.PRE	
FIA_UID.1.2	F.IA.UID F.IA.IDE F.IA.CSA F.IA.CON F.IA.CSN	F.IA.UID ensures that all users are uniquely identified for a database and F.IA.IDE ensures that the user's identity is established before any interaction. The other three security functions extend this requirement by ensuring that a user also has the correct privileges and security attributes to allow them access to that session.
FIA_ATD.1.1	F.IA.ATT	
FIA_USB.1.1	F.PRI.XVP	
FDP_ACC.1.1	F.OAC.SUA F.OAC.OBA F.OAC.POL F.OAC.OBID F.OAC.OBREF	The first three security functions define the DAC policy for the TOE, with F.OAC.OBID and F.OAC.OBREF providing support by ensuring uniqueness and correct referencing of object names.
FDP_ACF.1.1	F.OAC.OBA F.OAC.SUA F.OAC.OBID F.OAC.OBREF	

Table 3: Mapping of IT Security Functions to SFRs

SFR	IT Security Functions	Notes
FDP_ACF.1.2	F.OAC.POL F.OAC.OBID F.OAC.OBREF	F.OAC.POL defines the rules to determine if operations are allowed between controlled subjects and objects. F.OAC.OBID and F.OAC.OBREF provide support by ensuring uniqueness and correct referencing of object names.
FDP_RIP.1.1	F.OAC.OR	
FMT_MSA.1.1	F.IA.ATT, F.LIM.CNF, F.APR.GOP, F.APR.ROP, F.APR.GRSP, F.APR.GRR, F.IA.CNF	
FMT_MSA.3.1	F.OAC.POL	
FMT_MSA.3.2	F.OAC.POL, F.OAC.SUA, F.OAC.OBA	
FMT_MTD.1.1	F.AUD.ACC, F.AUD.DEL	
FPT_REV.1.1	F.APR.ROP F.APR.GRSP F.APR.GRR	Any privilege and role can be revoked from any user.
FPT_REV.1.2	F.PRI.SLOSE F.PRI.SINIT F.PRI.RPLOSE F.PRI.NULOSE	F.PRI.SLOSE defines the conditions under which a privilege will cease to become effective during a session. F.PRI.SINIT covers the conditions for effective privileges at session startup. F.PRI.RPLOSE covers the loss of privileges from roles and the PUBLIC user group and F.PRI.NULOSE covers the loss from normal users.
FMT_SMR.1.1	F.IA.CSA	
FMT_SMR.1.2	F.APR.DER, F.APR.EDR	

Table 3: Mapping of IT Security Functions to SFRs

SFR	IT Security Functions	Notes
FRU_RSA.1.1	F.LIM.RSESS F.LIM.RCALL F.LIM.TIME F.LIM.POL	This SFR has been split into three security functions, which separately cover session level limits, call level limits and timed limits. In addition, F.LIM.POL provides support by defining the enforcement rules concerning specific resource profiles and the default resource profile defined for the database.
FTA_MCS.1.1	F.LIM.NSESS	
FTA_MCS.1.2	F.LIM.NSESS	
FAU_GEN.1.1	F.AUD.SEV F.AUD.SOM F.AUD.ALW	These three security functions define the events for which audit records are created. F.AUD.SEV defines the main bulk or events recorded. F.AUD.SOM and F.AUD.ALW cover the specific cases of `CONNECT/DISCONNECT` events and events always recorded in the operating system audit trail.
FAU_GEN.1.2	F.AUD.INF	
FAU_GEN.2.1	F.AUD.INF	
FAU_SAR.1.1	F.AUD.VIEW	
FAU_SAR.1.2	F.AUD.ACC	There are two types of built-in views that can be used to view the audit data. There are those that can only be used by the administrator, and there are those that allow a normal user to view only those entries relating to themselves.
FAU_SAR.3.1	F.AUD.VIEW	
FAU_SEL.1.1	F.AUD.SEV	
FAU_SEL.1.2	F.AUD.SEV, F.AUD.CNF	

Table 3: Mapping of IT Security Functions to SFRs

SFR	IT Security Functions	Notes
FAU_STG.1.1	F.AUD.DEL	The audit trail can be written to two places - a database table or the operating system audit trail. The audit records stored in the database are permanent and can only be removed by a suitably authorised user, as described in F.AUD.DEL. Records written to the operating system audit trail (NT) can be configured to be permanent. This is configurable at the operating system level and not the database level.

Table 3: Mapping of IT Security Functions to SFRs

ANNEX

B ST Rationale

This section provides a demonstration of why the identified security objectives (chapter 4) are suitable to counter the identified threats and meet the stated security policies (chapter 3).

There are no additional threats or IT security objectives on top of those already contained within [CDBMS PP]. Neither are there any additional organisational security policies. Therefore see [CDBMS PP] for further information.

Security Requirements Rationale

Security Requirements Satisfy the Security Objectives

There are no additional IT security objectives. Therefore, the ST complies fully with CDBMS PP: see [CDBMS PP] for further information.

Assurance Requirements are Appropriate

The target assurance level is EAL4, which exceeds the assurance requirement of EAL3 as stated in [CDBMS PP]. No augmented assurance requirements are defined. See [CDBMS PP] for further information.

Security Requirements Dependency Analysis

There are no additional security requirements. Therefore, the ST complies fully with [CDBMS PP]: see [CDBMS PP] for further information.

Security Requirements Mutually Supportive

There are no additional security requirements. Therefore, the ST complies fully with [CDBMS PP]: see [CDBMS PP] for further information.

Security Functions Satisfy Security Requirements

This section demonstrates how the specified IT security functions satisfy all the SFRs

included in this ST. The following table shows how all the security requirements are satisfied by at least one IT security function and that every IT security function is used to satisfy at least one SFR.

	FIA				FDP				FMT							FRU	FTA		FAU							
	UID.1.1	UID.1.2	ATD.1.1	USB.1.1	ACC.1.1	ACF.1.1	ACF.1.2	RIP.1.1	MSA.1.1	MSA.3.1	MSA.3.2	MTD.1.1	REV.1.1	REV.1.2	SMR.1.1	SMR.1.2	RSA.1.1	MCS.1.1	MCS.1.2	GEN.1.1	GEN.1.2	GEN.2.1	SAR.1.1	SAR.1.2	SAR.3.1	SEL.1.1	SEL.1.2	STG.1.1
F.IA.PRE	Y																											
F.IA.UID		Y																										
F.IA.CNF									Y																			
F.IA.IDE		Y																										
F.IA.CSA		Y													Y													
F.IA.CON		Y																										
F.IA.CSN		Y																										
F.IA.ATT			Y																									
F.LIM.CNF																												
F.LIM.POL																	Y											
F.LIM.NSESS																		Y	Y									
F.LIM.TIME																	Y											
F.LIM.RSESS																	Y											
F.LIM.RCALL																	Y											
F.OAC.CON		Y																										
F.OAC.OBID					Y	Y	Y																					
F.OAC.OBREF					Y	Y	Y																					
F.OAC.SUA					Y	Y					Y																	
F.OAC.OBA					Y	Y					Y																	
F.OAC.POL					Y		Y			Y	Y																	
F.OAC.OR								Y																				
F.APR.GOP									Y																			
F.APR.ROP									Y				Y															
F.APR.GRSP									Y				Y															
F.APR.GRR									Y				Y															
F.APR.DER																Y												
F.APR.EDR																Y												
F.PRI.NULOSE														Y														
F.PRI.RPLOSE														Y														
F.PRI.SINIT														Y														
F.PRI.SLOSE														Y														
F.PRI.XVP				Y																								
F.AUD.SOM																			Y									
F.AUD.SEV																			Y									
F.AUD.ALW																			Y						Y	Y		
F.AUD.CNF																										Y		
F.AUD.ACC												Y												Y				
F.AUD.DEL												Y																Y
F.AUD.INF																				Y	Y							
F.AUD.VIEW																							Y		Y			

Table 4: Mapping of SFRs to IT Security Functions

Security Functions Mutually Supportive

The dependency analysis and demonstration of mutual support within [CDBMS PP] show how the SFRs are mutually supportive. Since this mutual analysis has already been demonstrated, this analysis focuses on any additional dependencies introduced as a result of the inclusion of additional detail within the IT security functions.

The following additional supportive dependencies exist between the identified security functions:

- F.PRI.SINIT together with F.PRI.SGAIN supports F.OAC.POL and F.IA.CSM by defining the conditions which will cause a privilege to be effective both during a

session and at the creation of a session.

- F.OAC.OBID and F.OAC.OBREF support F.OAC.SUA, F.OAC.OBA and F.OAC.POL by ensuring that object names are unique and that a particular object being referenced cannot be confused, deliberately or otherwise, with another object.
- F.LIM.POL provides support to F.LIM.RSESS, F.LIM.RCALL and F.LIM.TIME by defining the enforcement strategy for specified resource profiles and the default resource profile for the database.
- F.IA.CSA, F.IA.CON and F.IA.CSN provide support to F.IA.UID and F.IA.IDE by ensuring that a user is not only uniquely identified but also has the prerequisite privileges and security attributes for valid session creation.

Assurance Measures Rationale

The target assurance level is EAL4. No additional assurance measures are claimed.

This Page Intentionally Blank

ANNEX

C Mapping to ITSEC ST

This annex maps the components of this Security Target to the counterparts in [ITSEC ST].

Threats

The following table maps the threats in chapter 3 to [ITSEC ST]:

Threat	[ITSEC ST]	Comment
T.ACCESS	T.AXS.DB T.AXS.ANON, T.AXS.IMP	T.ACCESS is a generic threat covereing unauthorised access to the database. T.AXS.DB, T.AXS.ANON and T.AXS.IMP cover specific instances of T.ACCESS as: unauthorised access to the database, anonymous access and impersonation respectively.
T.DATA	T.AXS.GC, T.AXS.OBJ, T.AXS.RD	T.DATA is a generic threat covering unauthorised access to information within the database by an authorised database user. T.AXS.GC, T.AXS.OBJ and T.AXS.RD cover specific instances of access to database control data, controlled datrabase objects and residual data respectively.
T.RESOURCE	T.AXS.XCR	Identical.
T.ATTACK	T.REPEAT	T.ATTACK is an improvement of the wording of T.REPEAT, but the meaning is unchanged.
T.ABUSE	T.ABUSE	Wording changed slightly, but otherwise the meaning is unchanged.
T.OPERATE	T.OPERATE	
T.CRASH	T.CRASH	Identical.
T.BADMEDIA	T.BADMEDIA	Identical

Table 5: Threats

Threat	[ITSEC ST]	Comment
T.PHYSICAL	-	New threat not present in [ITSEC ST], note that since this threat is countered by the operarational environment, there are no implications for the scope of the evaluation.
T.TRUSTED	-	

Table 5: Threats

No threats in [ITSEC ST] have been omitted from the above table.

Secure Usage Assumptions

The following table maps the Secure usage assumptions in chapter 3 to [ITSEC ST]:

Assumption	[ITSEC ST]	Comment
A.OS	A.E.FC2, A.E.OSN	A.E.FC2 has no direct equivelent in this ST.
A.NETWORK	A.E.NETWORK	
A.PEER	-	Has no equivelent in ITSEC ST.
A.FILES	A.U.PTCB	A.U.PTCB has a more complete list of example files to which access should be restricted, however the intention is still the same.
A.APPLICA-TIONS	A.E.NETAPPS, A.E.CLIEN-TAPPS	
A.LOCATE	A.E.PAC	
A.PROTECT	A.E.PAC. A.E.PHYAC-CESS, A.E.IDOM, A.E.AUDM, A.E.MMED	
A.ACCESS	A.E.PAC, A.E.PHYACCESS	
A.MANAGE	A.U.TRUS, A.U.TRAUD	A.MANAGE is a more generic statement of system management requirements.

Table 6: Secure usage assumptions

Objectives

The following tables maps the security objectives in chapter 4 to [ITSEC ST]:

Objective	[ITSEC ST]	Comment
O.I&A	O.ACCESS.DB	The wording has changed, but the meaning is the same.

Table 7: Objectives

Objective	[ITSEC ST]	Comment
O.ACCESS	O.ACCESS	Unchanged.
O.ACCESS.DO	O.ACCESS.DO	Unchanged.
O.ACCESS.DA	O.ACCESS.DA	Unchanged.
O.ACCESS.DC	O.ACCESS.DC, O.ACCESS.AD	O.ACCESS.DC has been formed from two objectives in [ITSEC ST], O.ACCESS.DC covers database control data, wheras O.ACCESS.AD covers accountability data.
O.ACCESS.RESUSE	A.ACCESS.REUSE	Unchanged.
O.AUDIT	O.ACCOUNT	O.AUDIT is a more complete statement of the requirements tham O.ACCOUNT. Otherwise the objectives state similar intentions.
O.RESOURCE	O.ACCESS.GR	The wording has been changed, but the meaning is the same.
O.ADMIN	-	New objective.
O.INSTALL	A.U.EVAL, A.E.OSN	
O.PHYSICAL	A.E.NETWORK, A.E.PHYACCESS, A.E.IDOM, A.E.AUDM, A.E.MMED	
O.AUDITLOG	A.U.AUDA, A.E.AUDIT	
O.RECOVERY	A.E.SREC	
O.QUOTA	A.U.TSQA, A.U.PRIV	quotas = privileges (change??)
O.TRUST	A.U.TRUS, A.U.TRAUD	need to include the ALTER SESSION privilege.
O.AUTHDATA	A.E.AUTHD	
O.MEDIA	A.U.PMED, A.U.DMED, A.E.MMED	
O.CONFIGURE	A.U.SQL92, A.U.DBNAME, A.U.DNAM A.U.DBLK, A.U.DACC, A.U.SYS A.U.OSA, A.E.IDOM	DBNAME to include DACC is more restrictive

Table 7: Objectives

No objectives from [ITSEC ST] have not been included in the above table.

Of the environmental and intended method of use assumptions in [ITSEC ST], the fol-

lowing are not mapped in table 6 or table 7:

- A.U.REFINT, describes the operation of certain referential integrity mechanisms. As such it simply requires users to understand, what is written in the user documentation. It is felt that this does not make a material contribution to [ITSEC ST] or this document and therefore has beeen removed.
- A.E.FC2, is not appropriate in the context of a Common Criteria evaluation.

TOE Summary Specification

In general the IT Security Functions identified in are identical to the same named SEFs in [ITSEC ST], except as noted below:

- F.IA.PRE has been added to identify the operations that a user may perform prior to establishing a user session;
- F.IA.ATT has been added to expose the set of security attributes held for each user;
- F.OAC.ROL is a new SEF describing the operation of the ROLLBACK command;
- F.OAC.OR describes the operation of object re-use, it does not however introduce any new functionality;
- F.OAC.CON identifies a number of system configuration utilities required to install and configure the TOE;
- F.PRI.SINIT is an agregation of the [ITSEC ST] SEFs, F.PRI.NUGAIN, F.PRI.RPGAIN, F.PRI.SINIT & F.PRI.SGAIN.
- F.AUD.VIEW is a new SEF describing how the audit trail may be examined.

ANNEX

D References

[ITSEC] Information Technology Security Evaluation Criteria: Provisional Harmonised Criteria
Commission of the European Communities
Issue 1.2, 28 June 1991

[OR_ST7.0] Security Target for Oracle7 Database Server Release 7.0.13
Oracle Corporation
Version 4, 1993

[O7_DSV1] Guide to Distributed Systems, Volume 1
Oracle Corporation

[O7_ICG] Oracle7 for Microsoft Windows NT 3.51, Release 7.2 Installation and Configuration Guide

[O7_ROPFF] Required and Optional Platform Features and Functions
Oracle Corporation
Release 7.1.5

[O7_ROSDF] Required OSD Functions
Oracle Corporation
Release 7.1.5

[O7_SAG] Oracle7 Server Administrator's Guide
Oracle Corporation
Part Number A20322-2, April 1995

[O7_SADG] Oracle7 Server Application Developer's Guide, Release 7.2
Oracle Corporation
Part Number A20323-2, March 1995

[O7_SR] Oracle7 Server Reference, Release 7.2
Oracle Corporation
Part Number A20327-2, April 1995

[O7_SDA] Oracle7 Server Documentation Addendum Release 7.1
Oracle Corporation
Part Number A12042-3, 1994

[O7_SQLRM] Oracle7 Server SQL Language Reference Manual
Oracle Corporation
Part Number 778-70-1292, December 1992

[O7_SC] Oracle7 Server Concepts, Release 7.2
Oracle Corporation
Part Number A-20321-2, March 1995

[TCSEC] Trusted Computer Security Evaluation Criteria
DoD 5200.28-STD,
Department of Defense,
United States of America,
December 1985

[UKCB_WNT] Certification Report for the Microsoft Windows NT operating system software,
Version 3.51
UK ITSEC Scheme

[SIN 053] Scheme Information Notice No. 053,
F-C2 Functionality Class,
UK ITSEC Scheme,
Issue 1.0, 24 April 1996

[MEMO 1] CESG Computer Security Memorandum No. 1
Glossary of Computer Security Terms,
Issue 2.0, November 1989

[TDI] Trusted Database Management System Interpretation of the Trusted Computer System Evaluation Criteria, NCSC-TG-021, Version-1, April 1991

[CDBMS PP] Commercial Database Management System Protection Profile (CDBMS PP),
Issue 6.

[ITSEC ST] Security Target for Oracle 7 Database Server,
Release 7.2,
April 1997

ANNEX

E Glossary

Access

1) Condition where the potential exists for information to flow between subjects and objects. [Memo 1]

2) A specific type of interaction between a subject and an object which results in the flow of information from one to the other. [Memo 1]

In this Security Target, the term access is used in its broadest sense to refer to any operation, action or condition which allows or can result in the viewing, entry, modification and destruction of data and data objects, and/or consumption of resources.

Access Control

1) Control over the flow of information between subjects and objects. [Memo 1]

2) The prevention of access without access rights. [Memo 1]

Access Right

Authorisation of access [Memo 1]

Accounting

Recording of the exercise of access rights and other security relevant events to types of information specified (in the System Security Policy). [Memo 1]

Assertion

A explicit statement that security measures in an outer security domain constitute an adequate basis for security measures (or the lack of them) in an inner security domain. [Memo 1]

Asset

Information or resources to be protected by the technical and non-technical countermeasures of a TOE. [Memo 1]

Audit

1) An independent review and examination of system records and activities in order to test for the adequacy of system security measures, to identify the degree of conformance with established security policy and operational procedures and to recommend any indicated changes in measures, policy and/or procedures. [Memo 1]

2) Monitoring to detect and warn of events which might threaten security. [Memo 1]

Audit Trail — The set of records generated by a TOE in response to accountable operations, providing the basis for audit.

Authentication Data — Includes the user identifier, password and authorisations for each user of the product.

Authorisation — A right granted to a user to perform an action that would otherwise be prohibited by the product.

Availability

1) The prevention of the unauthorised withholding of information or resources. [Memo 1]
2) Continuous access to information or resources by authorised users. [Memo 1]

Call — A call to the Oracle7 Database Server OPI interface.

Command — An SQL command

Client — The process which requests services from a server.

Database link — A string that uniquely identifies to the communications software the location and name of the remote database.

Discretionary Access Control — Access control based on access rights granted by users other than the System Security Officer. [Memo 1]

Denial of Service

1) The unauthorised withholding of information or resources. [Memo 1]
2) The prevention of legitimate access. [Memo 1]

Evaluated Interface — A client-side facility which can be used to drive the server in a secure manner, but which is not a component of the TCB. Components which provide evaluated interfaces provide no SEFs in themselves, but have been tested by the evaluators to show that they contain no security vulnerabilities: e.g., they cannot be used to circumvent or corrupt the security enforcing components of the product.

Identification — The employment of user id etc. to enable computers to identify, and grant access to, authorised users. [Memo 1]

Instance — An instance of the Oracle7 Database Server is the set of Oracle background processes and allocated memory which perform the work of the product. An instance must be started up, that is the background processes created and the memory allocated, by an authorised user before the database which the instance manages can be accessed.

Least Privilege — The principle of granting only such access rights as are required for subjects to perform their authorised tasks. [Memo 1]

Named Object[1] — An object which may be written by a subject of one identity and read by a subject of a different identity and for which an explicit instance may be requested. For the TOE the named objects are database objects as defined in chapter 3, “Objects” on page 24.

1. This term and definition comes from [TCSEC] via [SIN 053]

Need-to-Know	Security principle that the dissemination of classified information should be no wider that is required for the efficient conduct of the business in hand and restricted to those who are authorised to have access. [Memo 1]
Node	A discrete computing element of a distributed computer system, which provides application services and/or database management services.
Object	A passive entity which contains or receives information.
Object Privilege	An access right to perform a specified operation on a specified Object-Privilege Object.
Object Reuse	Reuse of a storage medium for a different object. [Memo 1]
Operation	The execution of an SQL command
Platform	The combination of software and hardware layers underlying the DBMS.
Product	All hardware, firmware and software components that comprise the Oracle7 Database Server product.
Server	Process that provides one or more services, e.g. manages a resource such as a database.
Statement	An SQL Statement, consisting of a command and a number of clauses
Subject	An entity that causes information to flow among objects or that change the system status (subjects are represented on the system by processes). A subject is implemented at the operating system level by a Client or by a Server that is temporarily acting as a Client, and at the database level by a database user account that has an established database session.
System Privilege	An access right to perform a specified database operation or class of database operations across all schemas in a database.
User	Any individual/person who has a unique user identifier and who interacts with the Oracle7 Database Server product. A user who is not included under the trusted administrative role (e.g. the DBA user) is described as a normal user.
User Identification	A character string which uniquely identifies a user to a system. [Memo 1]

This Page Intentionally Blank