OLS Security Target
for Oracle9*i* Release 2 (9.2.0)

November 2002

Security Evaluations
Oracle Corporation
500 Oracle Parkway
Redwood Shores, CA 94065

OLS Security Target for Oracle9*i* Database Server
Release 2 (9.2.0)

November 2002

Authors: Daniel Elliott, Peter Goatly.

Contributors: Steve Hill, Shaun Lee, Paul Needham.

Contents

CHAPTER

1 Introduction

This document is the security target for the Common Criteria evaluation of Oracle9*i* Database Server, Release 2 (9.2.0) with Oracle Label Security.

Identification and CC Conformance

Title: OLS Security Target for Oracle9*i*

Target of Evaluation (TOE): Oracle9*i* Server Enterprise Edition, with Oracle Label Security.

Release: 9.2.0.1.0

Operating System Platforms: Microsoft Windows NT 4.0, Sun SPARC Solaris 8.

CC Conformance: Database Management System Protection Profile (DBMS PP) [DPP]. The authentication packages claimed for the Microsoft Windows platform are *OS Authentication* and *Database Authentication*. The authentication package claimed for the Sun Solaris platform is *Database Authentication*.
This Security Target conforms to [CC, Part 2] and [CC, Part 3]. All SFRs in the Security Target are derived from [CC]. ALC_FLR.3 is the only augmented assurance criterion specified.

Assurance: EAL4 augmented with ALC_FLR.3[1].

Keywords: Oracle9*i*, O-RDBMS, database, Oracle Label Security, OLS, security target, EAL4

Version of the Common Criteria [CC] used to produce this document: 2.1

1. ALC_FLR is specified in [CEM_FLR, Annex A], which provides replacement text for Clause 12.2 in Part 3 of [CC]. ALC_FLR.3 provides assurance at the highest defined component level that there are flaw remediation procedures for the TOE by which discovered security flaws can be reported to, tracked and corrected by the developer, and by which corrective actions can be issued to TOE users in a timely fashion.

TOE Overview

Oracle9*i* is an object-relational database management system (O-RDBMS), providing advanced security functionality for multi-user distributed database environments. The security functionality in Oracle9*i* includes:

- user identification and authentication, with password management options;
- discretionary access controls on database objects;
- granular privileges for the enforcement of least privilege;
- user-configurable roles for privilege management;
- extensive and flexible auditing options;
- secure access to remote Oracle databases, *and*
- stored procedures, triggers and security policies for user-defined access controls and auditing.

Oracle9*i* supports both client/server and standalone architectures. In addition, Oracle9*i* supports multi-tier architectures, however in this environment any tier (middle-tier) that communicates directly with the server is actually an Oracle client and any lower tiers are outside of the scope of this ST. In all architectures, the Oracle9*i* Server acts as a *data server,* providing access to the information stored in a database. Access requests are made via Oracle9*i interface products* that provide connectivity to the database and submit Structured Query Language (SQL) statements to the Oracle9*i* data server. The Oracle9*i* interface products may be used on the same computer as the data server, or they may run on separate client machines and communicate with the data server via network interfaces.

Oracle Label Security (OLS) enables application developers to add label-based access control (LBAC) to their Oracle9*i* applications. In addition to discretionary access control (DAC) that is provided by Oracle9*i*, it mediates access to rows in database tables based on a label (or labels) contained in each row, and the labels and privileges associated with each user session. Such labels quantify the sensitivity of data and the clearance of users to access sensitive data.

TOE Product Components

The Oracle9*i* Server Enterprise Edition with Oracle Label Security includes the products identified in Table 1. Access to the Oracle9*i* server is provided via the interface products identified in Table 2.

[OLS_ECD] defines which TOE products must be installed in the evaluated configuration and defines the requirements for setting up the TOE environment.

Table 1: TOE Server Products

TOE Server Products
Oracle9*i* Server Enterprise Edition 9.2.0
Oracle Label Security Release 9.2.0

Table 2: TOE Interface Products

TOE Interface Products
SQL*Plus 9.2.0
Oracle Call Interface 9.2.0
Oracle Net Services 9.2.0

Document Overview

This document consists of the contents of the OLS for Oracle8i Security Target, [OLS_ST8i], with appropriate additional material merged in to cover the new facilities. Change bars indicate the changes made relative to [OLS_ST8i].

Chapter 2 of this security target provides a high-level overview of the security features of the Oracle9*i* data server and Oracle Label Security. Chapter 3 identifies the assumptions, threats, and security policies of the TOE environment. Chapter 4 describes the security objectives for the TOE and for the environment needed to address the assumptions, threats, and security policies identified in Chapter 3. Chapter 5 identifies the Security Functional Requirements (SFRs), the Security Assurance Requirements (SARs) and the security requirements for the IT environment. Chapter 6 summarises each Security Function (SF) provided by Oracle9*i* and Oracle Label Security to meet the security requirements. Chapter 7 describes how the TOE conforms to the requirements of the DBMS Protection Profile and Chapter 8 provides the rationale for the security claims made within this security target.

Appendix A contains a list of references and Appendix B provides a glossary of the terms.

This Page Intentionally Blank

CHAPTER

2 TOE Description

This section describes the product features that provide security mechanisms and contribute to the security of a system configured using Oracle9*i* with Oracle Label Security. The security features of Oracle9*i* are explained primarily in [DAG, part IV] and in [CON, Part VII]. The additional security features provided by Oracle Label Security are explained primarily in [OLSAG, Part I]. In general, these descriptions correspond to the specifications of IT security functions provided in chapter 6 of this Security Target.

This chapter describes the major elements of the Oracle9*i* architecture, the types of database objects supported by Oracle9*i*, the access control mechanisms used to protect those objects, controls on user resource consumption, the accountability and auditing mechanisms, and the security management features provided by Oracle9*i*. The access control mechanisms consist of the discretionary access control supplied in Oracle9*i* together with the label-based access control supplied by Oracle Label Security. Additional Oracle9*i* security features that are not addressed by the security functional requirements of Chapter 5 are also briefly discussed.

Oracle9*i* Architecture

The Oracle9*i* architectural components are described in detail in [CON]. The additional components provided for Oracle Label Security are described in [OLSAG].

Database

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, views, etc.). In a distributed system there can be many databases: each database can contain many database objects, but each database object is stored within a single database.

Instance

An *instance* consists of a set of Oracle *background processes*, which do the work of the DBMS by executing Oracle9*i* software, and a shared memory area. 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.

Database Connections and Sessions

Each database user employs Oracle9*i* interface products to establish a *database connection* to an Oracle9*i server process* for a particular database instance. If the user is defined as a valid user for the database and has the required *privileges*, then the server will create a *database session* for the user. While connected, the user can make requests to the Oracle9*i* server to read and write information in the database. The server handles 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 privileges to database objects and other constraints configured by a *database administrative user*.

Distributed Databases

In a distributed environment, a user may access database objects from multiple databases. After establishing an initial database session on one instance, the user can transparently establish database sessions on other (remote) database instances using *database links*. A database link identifies a remote database and provides authentication information. By qualifying references to database objects with the name of a database link, a user can access remote database objects. However, each Oracle9*i* database instance is autonomous with respect to security — a remote server enforces security based on the privileges of the user as defined in that remote database.

Structured Query Language (SQL)

The Oracle9*i* server supports the ANSI/ISO SQL standard [SQL92] at the entry level of compliance and provides Oracle-specific SQL language extensions. All operations performed by the Oracle9*i* server 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 manipulate the data controlled by database objects in one of four ways: by querying the data held in a database object; by row insertions; by row deletion; by column update. They include the command to lock a database object.
- 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 statements dynamically manage the properties of a user's database session.
- System Control statements dynamically manage the processes and parameters of an Oracle9*i* database instance.
- Embedded SQL statements incorporate DDL, DML, and transaction control statements within a procedural language program.

Programming Language/SQL (PL/SQL) is a procedural language supported by Oracle9*i* that provides program flow control statements as well as SQL statements [PLS]. *Program units* written in PL/SQL can be stored in a database and executed during the processing of a user's SQL command.

Client side interfaces

The Oracle Call Interface (OCI - described in [OCI]) provides an application programming interface (API) for developing database applications written in high level languages such as C.

An Oracle9*i* Database

An Oracle9*i* database contains the data dictionary and two different types of database objects:

- schema objects that belong to a specific user *schema* and contain user-defined information [CON part IV]; *and*
- non-schema objects to organise, monitor, and control the database [CON part II], [DAG].

In an Oracle9*i* database there are two types of connections for users of the database:

- Administrator connection.
  This covers users who connect to the database via AS SYSOPER or AS SYSDBA by virtue of possessing either the SYSOPER or SYSDBA system privilege (see [DAG, 1-13]). Users making a connection AS SYSOPER are allowed to perform operator administrative tasks (e.g. database startup and shutdown, and ALTER DATABASE commands). Users making a connection AS SYSDBA are allowed to perform all administrative tasks (including granting and/or revoking object privileges on other users' objects);
- Normal connection (note that this includes users SYS and SYSTEM. [DAG, 1-12]).
  This covers users who are authorised to access the database by virtue of being explicitly defined and identified to an instance of the Oracle9*i* Database Server.

Note that connecting to a database via the CONNECT INTERNAL command is no longer supported in Oracle9*i*.

Data Dictionary

At the centre of an Oracle9*i* database is the data dictionary - a set of internal Oracle tables that contain all of the information the Oracle9*i* server needs to manage the database. The data dictionary tables are owned by the user SYS and can only be modified by highly privileged users. [CON] advises that no Oracle user should **ever** alter any object contained in the SYS schema and the security administrator should keep strict control of this central account. A set of read-only views is provided to display the contents of the internal tables in a meaningful way and also allow Oracle users to query the data dictionary without the need to access it directly.

All of the information about database objects is stored in the data dictionary and updated by the SQL DDL commands that create, alter, and drop database objects. Other SQL commands also insert, update, and delete information in the data dictionary in the course of their processing.

Schema Objects

A *schema* is a collection of user-defined database objects that are owned by a single database user. Oracle9*i* supports the schema object types identified in [SQL, 2-102].

A special schema PUBLIC is provided by Oracle9*i* to contain objects that are to be accessible to all users of the database. Typically, the kinds of objects that are created in the PUBLIC schema are:

- Public database links that define access to remote databases;
- Public synonyms which point to objects which all users may need to access.

Non-Schema Objects

[SQL, 2-103] lists object types that can be created and manipulated with SQL, but are not contained within a schema. These include tablespaces, roles, profiles and users.

The primary storage management database object is a tablespace — it is used to organise the logical storage of data. A suitably privileged user manages tablespaces to:

- create new tablespaces and allocate database files to the tablespace,
- add database files to existing tablespaces to increase storage capacity,
- assign default tablespaces to users for data storage, *and*
- alter tablespaces for backup and recovery operations.

Within the database files, Oracle9*i* allocates space for data in three hierarchical physical units: data blocks, extents, and segments. When a user creates a schema object to store data (e.g., a table), a segment is created and the space for the segment is allocated in a specific tablespace.

Database Users

Oracle9*i* has two kinds of user connection: administrative connection (connecting AS SYSDBA or AS SYSOPER) and normal connection. Throughout this document the following terms are used to classify the types of database users:

- Normal User/Database Subject:
  A user who is connected via a normal connection. Note that the pre-defined users SYS and SYSTEM can be normal users.
- Database Administrative User/Administrative User:
  Any user who is authorised to perform administrative tasks. This term covers:
    - A Normal User who is authorised to perform an administrative task via the possession of an administrative privilege which permits the operation of the task.
    - A user who connects to the database via an administrative connection. Users making an administrative connection are authorised to access the database by virtue of having the SYSDBA or SYSOPER system privilege (i.e. they possess OS platform specific access rights, or are listed in the Oracle9*i* password file as a SYSDBA or SYSOPER user).

Note that the word *authorised* is used (e.g. “an authorised administrative user”) to indicate that the user has the specific authorisation (e.g. via a privilege) for the operation under consideration.

Database security is managed by privileged users through the maintenance of users, roles, and profiles.

- USERS identify distinct database user names and their authentication method.
- ROLES provide a grouping mechanism for a set of privileges.

- PROFILES provide a set of properties (e.g., resource limits, password management options) that can be assigned to individual users.

Additional security can be provided via customised OLS security policies, each of which defines a set of labels and a set of rules that govern data access, based on these labels.

These security topics are discussed in detail in subsequent sections of this chapter.

Access Controls

Access control is the process of defining a user's ability to read or write information. Oracle9*i* always provides *discretionary access control* (DAC). When the Oracle Label Security (OLS) product has been installed, *label-based access control* (LBAC) can be applied in addition to DAC.

Discretionary Access Control

DAC can be used to selectively share database information with other users. This access control mechanism can be used to enforce need-to-know style confidentiality as well as control data disclosure, entry, modification, and destruction. In addition to the DAC controls enforced by the Oracle9*i* server, application-specific access controls can be implemented using views and triggers to mediate a user's access to application data.

The DAC mechanism controls access to database objects based on the privileges enabled in the database session. There are two types of DAC privileges: *object privileges* and *system privileges*. Both object and system privileges may be granted directly to individual users, or granted indirectly by granting the privilege to an Oracle *role* and then granting the role to the user. Privileges and roles may also be granted to PUBLIC, authorising all database users for the privilege. During a database session, the privileges enabled in the session may be changed using several Oracle9*i* mechanisms that affect the set of privileges held by the session.

System Privileges

Oracle9*i* provides over 80 distinct system privileges to support the concept of least privilege — each database user can be granted only those system privileges that are needed to perform his or her job function. Often end-users would only need a minimal set of system privileges to connect to the database. Some users may be granted more powerful system privileges to authorise them to manage administrative objects, bypass particular server access controls, or perform specialised operations. A user may grant a system privilege to additional database users only if he or she holds that privilege with an administrative option (WITH ADMIN OPTION).

Object Privileges

An object privilege is permission to access a schema object in a prescribed manner (e.g., to INSERT rows into a table or EXECUTE a stored procedure). The owner of the schema containing the object may grant object privileges to other database users or roles. In addition, the owner may grant other users the right to grant those object privileges to additional database users (WITH GRANT OPTION).

Because object privileges are granted to users at the discretion of other users, this type of security is termed discretionary. Oracle9*i* ensures that users who attempt to gain access to objects have been granted the necessary object privileges for the specific operation, or have an overriding system privilege or role. The owner of an object always has total access to that object.

Roles

Oracle9*i* 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,
- Privilege bracketing, *and*
- Consistency.

Reduced privilege administration

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

The roles granted to a user can be selectively enabled (available for use) or disabled (not available for use). This helps a user to control use of those privileges which could result in unintended disclosure, entry, modification, or destruction of data.

Privilege Bracketing

Because the Oracle data dictionary records which roles have been granted to 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.

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 Application Roles

A secure application role is a role which is enabled by a PL/SQL package. A database administrative user can grant a secure application role all the privileges necessary to run a particular application. The role will then only be enabled if the application's check of the relevant conditions is successful. This means that the use of such a role can be based on information about the user's session, such as the IP address of a user who has connected through a proxy.

DDL Restriction

Privileges held via roles cannot be used with DDL statements that require access to database objects. For example, to create a view, a user requires access to the tables referenced by the view. The user must have *directly granted privileges* authorising the access to the underlying tables. Privileges held via a role are not applicable when the server performs the object access checking on DDL statements.

Pre-defined Roles

By default Oracle9*i* databases contain several pre-defined roles including:

- CONNECT — containing the system privileges to connect and create basic schema objects,
- RESOURCE — containing the system privileges necessary to create PL/SQL program units and triggers, and
- DBA — containing all system privileges WITH ADMIN OPTION.

These roles are provided for backward compatibility [DAG, 25-5] and can be modified or removed by suitably privileged users.

Session Privileges

During the database session, the privileges held by the session can vary. When a database session is initially established, it has all of the system and object privileges directly granted to the user in addition to those granted to PUBLIC. The session also has all of the privileges granted to any default roles associated with the user. The set of privileges can be changed by:

- Enabling and disabling roles,
- Accessing a view,
- Executing a stored program unit, *or*
- Firing a trigger.

Enabling Roles

During a database session, a user can enable and disable any granted role. Consequently, the privileges of the database subject can be modified to reflect different requirements for access to database objects.

Views

When a user creates a view, that user must have directly granted privileges that authorise access to all of the tables (or views) referenced in the view's query. In addition, if the user holds the necessary privileges WITH GRANT option or WITH ADMIN option, then the user may grant access to the view to other database users, authorising them for indirect access to the tables in the view. In this way, views can be used to restrict access to information based on complex SQL queries that select only the authorised data from the tables.

Stored Program Units

In order to use a stored program unit (procedure, function, or package), a user must have the privilege to EXECUTE the program unit. However, when the program unit runs, the privileges for its execution may be set to the owner's directly granted privileges (definers rights), or the invoker's privileges (invokers rights) depending on options set when the program unit is created. This allows access privileges to be encapsulated with the database operations being performed by the program unit. Any user with EXECUTE privilege for the program unit is authorised to indirectly access any database objects accessible to the program unit's owner.

Information about stored program units which have policy privileges for Label-Based Access Control is given in the section on "Trusted Stored Program Units" below.

Triggers

The security context for the execution of triggers is similar to that of stored program units. When a trigger fires as a result of a table access, the execution privileges for the trigger are set to the trigger owner's directly granted privileges rather than the privileges of the user who initiated the table update.

Information about labels and policy privileges for Label-Based Access Control for triggers is given in the section on "LBAC and Triggers" below.

Fine-grained Access Control

Fine-grained (or row-level) access control is available with the virtual private database (VPD) technology which is a standard feature of the Oracle9*i* Enterprise Edition. Fine-grained access control allows a database administrative user to associate security policies with tables, views and synonyms. These policies are implemented by PL/SQL functions and are enforced on a normal user no matter how the data is accessed (unless the user is authorised by the possession of the system privilege EXEMPT ACCESS POLICY).

Different policies can be applied for SELECT, INSERT, UPDATE and DELETE operations. Note that the use of the Oracle9*i* MERGE SQL command causes SELECT and INSERT or UPDATE operations to be performed. Note also that it is possible for more than one policy to be applied to a table, including building on top of base policies in packaged applications.

Application Context

An application context allows an application to make security decisions based on additional attributes attached to a user's session information. An application context provides a protected session persistent storage area for additional user attributes defined by the application.

To support application managed session pooling by middle tier applications, the DBMS_SESSION interface for managing application context is enhanced for Oracle9*i*. This interface now has a client identifier for each application context so that the application context can be managed globally while each client will see only their assigned application context.

Partitioned Fine-grained Access Control

Oracle9*i* provides the ability to partition security policy enforcement by application. This enables different security policies to be applied, depending upon which application is accessing the data. Oracle9*i* enables partitioning of fine-grained access control through policy groups and a driving application context. The driving application context securely determines which application is accessing the data, and policy groups facilitate the management of policies which apply by application.

A database administrative user specifies which policy group the policy falls into when adding a policy to a table/view using the ADD_GROUPED_POLICY interface. The driving context is defined using the ADD_POLICY_CONTEXT interface.

Label-Based Access Control

OLS provides label-based access control, which builds on VPD to mediate access to data at a row level without any code having to be written. Each data row is given one or more labels, each of which is used to store information about data sensitivity.

To be allowed access to a row, a user must satisfy both OLS label-based access control (LBAC) and Oracle9*i* DAC requirements which are based on the user's system-level privileges and database object privileges. Thus, to gain access to a row, a user must first be authenticated to the Oracle9*i* database. Second, the user must have the DAC object and system privileges required for the operation. Finally, the user must meet the criteria enforced by LBAC, which are based on the labels of the user and the data row.

In most applications, a relatively small number of application tables will require label-based access controls, while the protection provided by standard DAC will suffice for the majority of tables.

Data Labels

In OLS, each row of a table can be labelled as to its level of confidentiality. Each label contains three components:

- a single hierarchical level or sensitivity ranking,
- one or more horizontal compartments or categories, and
- one or more hierarchical groups.

The level specifies the sensitivity of the data. A typical organisation might define levels UNCLASSIFIED, CONFIDENTIAL, SENSITIVE, and HIGHLY_SENSITIVE. Alternatively, a commercial organisation might define levels only for PUBLIC and COMPANY_CONFIDENTIAL data.

The compartment component is non-hierarchical; compartments are typically defined to segregate data - such as data related to a particular ongoing strategic initiative. For example, a commercial organisation might define compartments for FINANCIAL, OPERATIONAL, SECURITY and PERSONNEL data.

Finally, groups are used to record ownership and can be used hierarchically. For example, FINANCE, SALES and ENGINEERING groups can be defined as children of a CORPORATION group, creating an ownership relation. In this example, the FINANCE, SALES and ENGINEERING groups are conceptually part of the CORPORATION group and any user authorised to access data which has a label that contains the CORPORATION group will also be authorised to access data which has a label containing one or more of the FINANCE, SALES or ENGINEERING groups.

Labels can contain a single level component, a level combined with a set of either compartments or groups, or a level and both compartments and groups.

OLS Administrators

There are two main roles for users involved in administering Oracle Label Security for a database: the LBAC Administrator role and OLS Policy Administrator roles. Throughout this document the following terms are used to describe these users:

- LBAC Administrator: A user who is able to create, alter and drop OLS policies in the database by virtue of possessing the LBAC_DBA role and `EXECUTE` privilege on the SA_SYSDBA package;
- OLS Policy Administrator / Policy administrator: A user who is able to execute the administrative packages for the OLS policy for which they possess the corresponding *policy*_DBA role. This user should be granted the EXECUTE privilege only on the OLS administrative packages that they require for their role.

Each OLS policy must have at least one OLS policy administrator. The same person could be the administrator for more than one policy.

Label Authorisations

A Policy Administrator can grant to users label authorisations which determine what kind of access (read or write) they have to the rows that are labelled. These authorisations are explained further in the sections below.

Session Label

Each OLS user has *user label authorisations* which are stored in the data dictionary and include:

- a maximum and minimum level,
- a set of authorised compartments,
- a set of authorised groups, and
- for each compartment and group, a specification of read-only access, or read-write access.

When the Policy Administrator sets up the user label authorisations for the user, he or she also specifies the user's initial session label.

The session label is the particular combination of level, compartments, and groups at which a user works at any given time. The user can change the session label provided that it remains within the user's label authorisations.

Row Label

When the Policy Administrator sets up a user's label authorisations, he or she also specifies an initial default row label which is used when a session is started up.

The row label is the particular default label assigned to data which a user enters during a session (if the user is not permitted to define the label explicitly). It can be changed

by the user to any level, from the one specified in the user's current session label, down to the user's minimum level. It can include only compartments and groups contained in the current session label, and for which the user has write access.

OLS Policies

OLS policies are established to specify how label-based access control is to be enforced on a database. Each OLS policy is created by an LBAC Administrator and the Policy Administrator then defines a set of labels and a set of enforcement options to govern LBAC access to data. These enforcement options provide for maximum flexibility in controlling the different Data Manipulation Language operations that users can perform. For each operation - SELECT, INSERT, UPDATE, and DELETE - administrators can specify a particular type of enforcement of the security policy. Note that the use of the Oracle9*i* MERGE SQL command causes SELECT and INSERT or UPDATE operations to be performed.

One or more policies can be applied to each table. A policy can also be applied to a schema. This has the effect of applying the policy to each table contained within the schema. Each row in each table in the database has a label column for each policy that applies to the table. For each OLS policy, Policy Administrators give user label authorisations to users and assign policy privileges to users and stored program units to permit access to data in tables controlled by the policy.

Policy Privileges

Policy privileges enable a user or stored program unit to bypass aspects of the label-based access control policy. In addition, the Policy Administrator can authorise the user or program unit to perform specific actions, such as the ability of one user to assume the authorisations of a different user.

Policy privileges can be granted to program units to authorise the procedure rather than the user to perform privileged operations. When only stored program units, and not individual users, have policy privileges, the system is most secure. Further, such program units encapsulate the OLS policy, which minimises the amount of application code that needs to be reviewed for security.

OLS Administration Tools

OLS provides administrative interfaces via packages supplied with OLS to define and manage OLS policies for a database. Initially, an LBAC Administrator must create a policy and then a Policy Administrator defines the levels, compartments, and groups that compose the labels, and then she or he can define the set of valid data labels for the policy.

The Policy Administrator can then use the administrative interfaces to:

- set the policy enforcement options,
- apply the policy to tables and schemas,
- authorise users,
- assign privileges to users and stored program units, and
- configure auditing.

The Oracle Policy Manager is a graphical user interface which can be used to call the OLS packages to perform the administrative functions for OLS policies. This GUI tool is not part of the TOE.

Relationships between Labels

When checking whether a user can read labelled data, OLS uses the dominance relationship between two labels. Provided that the policy enforcement option INVERSE_GROUPS is not in operation, if Label1 and Label2 are such that:

- Label1's level is greater than or equal to Label2's level, and

- Label2 contains one or more groups and Label1 contains at least one of the groups which belong to Label2 (or the parent group of one such subgroup), and
- Label1 contains all the compartments which belong to Label2,

then Label1 is said to "dominate" Label2.

If the policy enforcement option INVERSE_GROUPS is in operation, then [OLSAG, 13] defines a different dominance relationship for labels.

If a user's label dominates the label of a data item, then OLS allows the user to read that item (provided that the DAC rules also permit the user to access the data item).

Label Functions

OLS provides functions and procedures to manipulate labels. These include:

- functions to determine whether, given two labels, one label dominates the other or the labels are not comparable,
- functions to find the least upper bound and the greatest lower bound of two or more labels,
- a function to merge two labels together,
- a procedure to set the label of the current database session,
- a procedure to set the default row value for the current database session,
- a procedure to restore the label and the default row value for the current database session,
- a function to return the security attributes of the current database session.

Trusted Stored Program Units

Stored program units can become "trusted" when a Policy Administrator assigns them policy privileges. A stored program unit can be run with its own autonomous policy privileges, rather than those of the user who invokes it. For example, if a user possess no policy privileges, but executes a stored program unit which has the WRITEDOWN privilege, the user can update labels. In this case, the policy privileges used are those of the stored program unit, and not the user's. Trusted program units can encapsulate privileged operations in a controlled manner. By using procedures, packages, and/or functions that have been assigned policy privileges, a user may be able to access data that his or her own labels and policy privileges would not authorise. For example, to perform aggregate functions over all of the data in a table, not just the data visible to the user, a user could make use of a trusted program unit set up by an administrator. Program units can thus perform operations on behalf of users, without the need to grant policy privileges directly to users.

LBAC and Triggers

When a trigger fires, it is executed with the session label and with the policy privileges of the user that invoked the trigger.

Flashback Query

This Oracle9*i* feature allows data to be queried from a point in the past. Once a user has set the date and time that they would like to view, any SQL query that they execute will operate on data as it existed at that point in time. This can allow suitably authorised users to correct their own mistakes. SQL operations can be used to view the change history in order to identify the error. The error can then be backed out of by restoring data as it existed before the error.

To use Flashback, a user that possesses the ALTER SYSTEM privilege must first set

the undo retention interval (in seconds) long enough to be able to reconstruct the data.

Users who are to be permitted to use the Flashback feature must be granted the EXECUTE privilege on the DBMS_FLASHBACK package. This is the package which implements the feature.

While DBMS_FLASHBACK is enabled, read-only operations can be performed. For discretionary access control (DAC), access mediation for SQL queries performed on data at the specified point in the past will use the current DAC security attributes. For label-based access control (LBAC), the LBAC attributes of the current session will be used along with the label column values of the past data.

Note that the Flashback functionality does not reverse certain DDL statements such as DROP or TRUNCATE commands. It also does not apply to packages, procedures, or functions.

Quotas

Using Oracle9*i* profiles, a database administrative user can set quotas on the amount of processing resources a user can consume during a databases session. Limits can be specified for the following:

- enabled roles per session (via an init.ora parameter)
- database sessions per user,
- CPU time per session,
- CPU time per SQL call,
- connect time per session,
- idle time per session,
- database reads per session,
- database reads per SQL command, *and*
- a composite limit (based on CPU time, connect time, and database reads).

Once a profile has been created, it can be assigned to one or more users, depending on their need for processing resources. When a user exceeds the resource limit, the Oracle9*i* server will abort the operation, and, in some cases, terminate the user's session, or, in other cases, simply terminate the current SQL statement or rollback the current transaction.

A database administrative user may also set quotas on the amount of storage space that can be allocated for each user's schema objects in any specific tablespace.

Resumable statements are a feature in Oracle9*i* which allows an administrator to temporarily suspend a large operation, such as a batch update data load. This might be necessary when space has run out. Suspending the operation gives the database administrator an opportunity to take corrective steps to resolve the error condition. After the error has been corrected, the suspended operation automatically resumes execution. A suspended resumable operation is aborted automatically if the error is not fixed within a set time period.

Users must have the RESUMABLE system privilege before they can execute resumable operations. An ALTER SESSION ENABLE RESUMABLE statement is provid-

ed to enable SQL statements to be resumable when they are invoked within the session. Resumable operations are suspended under one of the conditions: Out of space, Space limit error, or Space quota error.

Identification and Authentication

Oracle9*i* always identifies authorised users of an Oracle9*i* database prior to establishing a database session for the user. Authentication can be performed directly by the Oracle9*i* server using passwords managed by the server, or the server can rely on the authentication done by the underlying OS platform.

For OS authentication, the database user connects to the Oracle9*i* server without specifying a user name or password. The server obtains the user's identity from the OS, and if the user is an authorised database user, a database session is created.

For Oracle authentication, a user must specify a user name and password in order to connect. The password is compared to the password for the user stored in the data dictionary and if they match, a database session is created. The user's password is stored in the data dictionary in a one-way encrypted form, so before the comparison is made, the password specified by the user is also one-way encrypted.

Password Management

A user may change his or her password at any time. Oracle9*i* provides the facility for suitably privileged users to create password complexity check functions that can screen new passwords for certain criteria, e.g.:

- a minimum number of characters in length;
- not equal to the user name;
- includes a minimum number of alphabetic, numeric, or punctuation characters;
- does not match any word on an internal list of words;
- differs from the previous password by a certain number of characters.

A suitably authorised user can also set password lifetime, a failed logon count leading to account lockout, expiration options, and password reuse requirements in an Oracle9*i* profile. By assigning different profiles to different groups of users, the password management parameters can vary among users.

By default the database does not enforce any password profile limits, however it is critical that certain password controls are used in all profiles such that the TOE achieves a *high* strength of function for the password mechanism (see the Minimum Strength of Function section in chapter 5). Guidance covering the different password controls, and instructions for modifying profiles to achieve SOF-*high*, is provided in the TOE's Evaluated Configuration Document [OLS_ECD].

Special Authentication

Database administrative users may connect to the database to perform functions such as starting up or shutting down an Oracle9*i* instance. These users can be authorised by either the use of a password file, or by having platform-specific access rights.

Platform-specific access rights are normally established by being a member of a special operating system group. On a UNIX platform, the group defaults to the 'dba' group but can be changed. On a Windows NT platform, the fixed group is ORA<SID>_OPER or ORA<SID>_DBA.

When a database administrative user wants to undertake special operations, he or she connects to the database through a special keyword: AS SYSDBA or AS SYSOPER. When connected using the AS SYSDBA keywords the database session then runs as the user SYS. When connected using the AS SYSOPER keyword the database session then runs as the user PUBLIC.

Auditing

Oracle9*i* ensures that relevant information about operations performed by 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 or her actions. Oracle9*i* does this by providing auditing options which are designed to be as granular and flexible as possible to ensure that exactly what needs to be 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. Oracle9*i* provides capabilities to permit auditing plans to be quickly enabled to implement crisis responses. In addition to the standard Oracle9*i* auditing features described here, application-specific audit trails can be implemented using triggers to capture auditing details about the changes made to the information in the database.

Audit Categories

A database administrative user can request auditing of a 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 specific 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.

Audit Options

Database administrative 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 events, 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 database session; by access results in a record for every occurrence of an audited action.

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

Fine-Grained Auditing

Database administrative users can request fine-grained auditing to monitor query access based on content. Whenever the policy conditions are met for returning a row from a query block, the query is audited. These policies are implemented by PL/SQL functions.

Fine-grained auditing is supported only with cost-based optimisation.

Audit Records

Oracle auditing permits audit information to be written to a database audit trail or to the audit trail of the underlying operating system. Audit records always include 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.

Audit Analysis

If Oracle writes to the database audit trail, then the powerful SQL data manipulation facilities of the DBMS can be used by database administrative 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 views on the database audit trail to assist in such audit analysis.

If Oracle is configured to write to an operating system 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 an Oracle database for comprehensive audit analysis using the SQL data manipulation facilities of the DBMS.

Auditing of SYS

Connections AS SYSDBA and AS SYSOPER along with attempts to startup or shutdown an instance are always recorded in the OS platform audit trail because they are OS events and because the database may not be available to be written into.

Oracle9*i* provides for information to be written to the OS platform audit trail about all SQL commands performed by users connected as the special user SYS and users connected through the keywords AS SYSDBA and AS SYSOPER. Such OS audit trail files should have OS DAC protection set by the OS system administrator to prevent all database users being able to tamper with them (including those users who are able to connect to the database as the special user SYS or through the keywords AS SYSDBA or AS SYSOPER).

Additional Auditing for OLS

OLS auditing supplements standard Oracle auditing by tracking use of its own administrative operations, and use of the policy privileges. Administrators can use either the SA_AUDIT_ADMIN package or Oracle Policy Manager to set and change the auditing options for an OLS policy.

When administrators create a new OLS policy, a label column for that policy is added to the database audit trail. The label column is created regardless of whether auditing is enabled or disabled, and independent of whether database auditing or operating system auditing is used. Whenever a record is written to the audit table, each policy provides a label for that record to indicate the session label. The label column is hidden (and hence cannot be explicitly selected by the user), but the administrator can create

audit views to display these labels. Note that in the audit table, the label does not control access to the row; instead, it simply records the sensitivity of the row.

The auditing options which administrators specify apply only to subsequent sessions, not to the current session.

Notes:

- All audit records for OLS events are written directly to the database audit trail, even if operating system auditing is enabled.
- If auditing is disabled, then no OLS audit records are generated.
- Labels are not present in audit data written to the operating system audit trail.
- The audit trail is held in a table called AUD$, which is moved from the SYS schema to the SYSTEM schema when OLS is installed.

Security Management

Oracle9*i* provides a number of mechanisms to support the management of database security. This section discusses the administrative system privileges, the importance of the initialisation file, the use of AS SYSOPER and AS SYSDBA, and Oracle9*i* server dependencies on the administration of the underlying OS platform.

Administrative Privileges

Oracle9*i* contains over 80 distinct system privileges. Each system privilege allows a user to perform a particular database operation or class of database operations. If a user has no privileges then they cannot perform any operations, including connecting to the database.

Database Administrative Users acquire the ability to perform administrative functions by being granted specific administrative system privileges. Other users are given only a minimal set of privileges allowing them to connect to the database and access the necessary data.

Oracle9*i* security management can be delegated to any number of users. Site-specific roles can be defined to delegate administrative responsibilities based on organisational structures.

Initialisation File

When an Oracle9*i* instance is started, the parameters specified in an initialisation file specify operational characteristics of Oracle9*i* server functionality, including security functionality. It is critical that the security parameters specified in the initialisation file for the instance be set to the values which conform to the evaluated configuration. The parameter values required by this security target are identified in the TOE's Evaluated Configuration Document [OLS_ECD].

SYSDBA and SYSOPER

When a user is connected AS SYSOPER or AS SYSDBA, the user is authorised to perform special database operations. Authorisation to connect as AS SYSDBA or AS SYSOPER is made via OS mechanisms (i.e., membership in an OS-defined group and requires that a user be authenticated by the OS), or by an Oracle9*i* password.

A user connected AS SYSOPER is authorised to perform database startup, shutdown, create server parameter file and backup operations. A user connected via AS SYSDBA has the same authorisations as SYSOPER with the additional capabilities to create databases and perform the operations allowed by all system privileges WITH ADMIN

option. Users who connect via AS SYSDBA have access to all of the data dictionary tables and can grant and/or revoke object privileges on other users' objects.

OS Administration

The security of the data managed by the Oracle9*i* data server is dependent not only on the secure administration of Oracle9*i*, but also on the correct administration of the underlying OS platform and any other nodes connected in a distributed environment. The requirements on OS and network configuration for this security target are identified in the TOE's Evaluated Configuration Document [OLS_ECD]. Guidance on the correct configuration of Oracle9*i* for a specific OS platform is contained in the *Oracle9i Installation and Configuration Guide* [ICG] for that platform. Finally, *Oracle Label Security Installation Notes* [OLS_IN] defines additional OS settings that are necessary when installing OLS.

Secure Distributed Processing

The basic distributed features included in the Oracle9*i* server make use of database links to define a connection path to a remote Oracle database. When a connection is made to a remote database, the information in the database link definition is used to provide identification and authentication information to the remote Oracle server. The remote server creates a database session for the user specified by the database link (if the user is authorised for access to the remote database) and then makes its access control decisions based on that identity and its privileges *in the remote database*.

By using database links to qualify schema object names, a user in a local database can

- select (e.g., join) data from tables in any number of remote Oracle databases,
- use DML statements to update tables in remote Oracle databases (Oracle9*i* automatically implements a two-phase commit protocol), and
- execute stored program units in remote Oracle databases.

Access to the remote database is transparent; however careful administration and control of the distributed environment is essential (see [DAG, 29: Managing a Distributed Database]). Access to non-Oracle distributed databases is provided by Oracle9*i*, but such databases are not part of the evaluated configuration.

OLS supports distributed operation when labels in the local and remote databases are compatible. Distributed databases behave in the standard way with OLS: the local user ends up connected as a particular remote user. OLS protects the labelled data, whether the user connects locally or remotely. If the remote user has the appropriate labels, he or she can access the data. If not, then access will be prevented.

Other Oracle9*i* Security Features

In addition to the security features described above, Oracle9*i* provides features which are related to security but do not directly address any of the functional requirements identified in this Oracle9*i* Security Target. These features provide significant security capabilities to support robust and reliable database applications. Apart from Data Integrity, for which no specific security functionality is claimed in Chapter 6, the features described below are not part of the evaluated configuration defined in [OLS_ECD].

Data Integrity

Oracle9*i* 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 tables are properly maintained.

Import/Export

It is important to ensure that data can be moved out of one database and re-inserted into the same or a different database while maintaining the data integrity and confidentiality. 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.

When a database object is exported, the list of users having object privileges to access the object can also be exported and then imported into the new database with the database object.

When tables protected by label-based access controls (LBAC) are exported via OLS, their label columns and the applied policies are also exported automatically.

Backup and Recovery

Backup of an Oracle9*i* database can be performed using platform-specific backup programs, the Oracle9*i* import/export utilities, or the Oracle9*i* recovery manager. The choice of mechanism depends upon the application needs, but all approaches can provide secure, reliable backup and recovery of the database.

The Oracle9*i* transaction integrity mechanisms also provide the basis for secure recovery following the failure of an Oracle9*i* instance or platform operating system. Whenever an Oracle9*i* instance is started, any transactions that were not committed prior to the failure are rolled back. This returns all of the information in the database, including the data dictionary tables, to a consistent and secure state.

Oracle Advanced Security

Oracle Advanced Security is an optional product which provides encryption of the Oracle network traffic between clients and servers and between two communicating servers and adaptors for various external authentication services and certificate authorities. The Oracle Internet Directory is a further add-on product that supports global authentication and global management of Oracle roles.

Supplied Packages

A number of standard packages are available to install in an Oracle9*i* database. These provide supportive functionality that can be invoked by other users and applications. They provide the following types of functions:

- Access to SQL features from PL/SQL programs, including dynamic SQL,
- Alert mechanisms for asynchronous notification of database events,
- File access functions to read and write OS files,
- Job queues for scheduling repeating administrative procedures,

- Lock management functions for user-defined locks,
- Oracle pipes for communication among database sessions,
- Output operations for procedure debugging,
- Functions to manipulate LOBs,
- Queues for asynchronous message generation and delivery (Advanced Queuing),
- Administration of distributed transactions and snapshots, and
- HTTP callouts to access Web services.

Oracle Policy Manager

A set of standard packages is provided when OLS is installed. They implement the majority of OLS's facilities. Administrators may choose to use these packages via the Oracle Policy Manager GUI rather than by making direct calls.

External Authentication Services

In addition to the standard Oracle9*i* database authentication and OS authentication methods described above, Oracle9*i* can be configured to use an external third party authentication service.

Application-Specific Security

Roles can be protected by use of a password. Applications can be created specifically to enable a role when the application is supplied with the correct password. Users can not enable the database role if they do not know the password.

Support for SQLJ

SQLJ allows application programmers to embed static SQL operations in Java code in a way that is compatible with the Java design philosophy. Oracle provides support for SQLJ at both the client and server, so that database applications written in Java may be executed at the client or at the server.

Oracle supports two SQLJ client side models; a thick client model where Java programs can make calls to the database via OCI using Oracle Net Services, and a thin client model where Java programs can call the database server directly bypassing the Oracle Net Services interface.

This Page Intentionally Blank

CHAPTER

3 Security Environment

Threats

As per [DPP, 3.2] with the following addition:

Threats countered by the TOE

T.LBAC *Unauthorised Access to Labelled Information.* An authorised database user accesses labelled information contained within a database without having the authorisation to access that information.

Organisational Security Policies

As per [DPP, 3.3] with the following additions:

P.LABEL Labels can be associated with subjects and with storage objects which are rows within tables:

a) A label is composed of an hierarchic level (classification), a set of non-hierarchic categories, and a set of hierarchic groups, as determined by the organisation who owns the information stored in the database.

b) A storage object label reflects the sensitivity of the information stored in the object.

c) A subject label reflects the authorisation of the subject to access the organisation's labelled information according to defined access rules.

P.INFOFLOW Information flow from entity A to entity B shall be permitted only if it does not result in a subject being able to observe labelled information that the subject is not authorised to see.

Assumptions

As per [DPP, 3.4] with the following modifications and additions:

TOE Assumptions

A.TOE.CONFIG The TOE is installed, configured and managed in accordance with [OLS_ECD], its evaluated configuration.

Note that [DPP, 3.4.2.2] includes assumptions about the secure configuration of the operating system underlying the TOE. In particular, A.ACCESS requires that the underlying system is configured such that only the approved group of individuals may obtain access to the system. [OLS_ECD] describes how the TOE and the system underlying it must be configured for the TOE to be in its evaluated configuration. This includes only allowing administrators to logon to the TOE's underlying operating system.

Underlying System Assumptions

A.MIDTIER To ensure accountability in multi-tier environments, any middle-tier(s) will pass the original client ID through to the TOE.

Personnel Assumptions

A.USERS Users are assigned label authorisations and policy privileges commensurate with the degree of trust placed in them by the organisation that owns, or is responsible for, the information processed by or stored in the TOE.

CHAPTER

4 Security Objectives

TOE Security Objectives

As per [DPP, 4.1] with the following addition:

O.ACCESS.LBAC The TOE must provide the ability for labels to be associated with subjects and database objects in accordance with the P.LABEL security policy. For entities which have been associated with labels, the TOE must use these labels as a basis for implementing an information flow control policy in accordance with the P.INFOFLOW policy.

Environmental Security Objectives

As per [DPP, 4.2] with the following addition:

O.USERS Those responsible for the TOE must ensure that users are assigned label authorisations and policy privileges commensurate with the degree of trust placed in them by the organisation that owns, or is responsible for, the information processed by or stored in the TOE.

This Page Intentionally Blank

CHAPTER

5 IT Security Requirements

TOE Security Functional Requirements

Table 3 below lists each Security Functional Requirement (SFR) included in this Security Target. SFRs in this table that are not included in [DPP] relate to requirements for label-based access control functions and are indicated by a "*" after the component identifier. Table 3 identifies which Common Criteria operations (assignment (A), selection (S), refinement (R), and/or iteration (I)) have been applied to the requirement relative to the DBMS Protection Profile [DPP] or relative to Part 2 of [CC] (for SFRs that are not in [DPP]).

The remainder of this section details the functional requirements as completed for this Security Target. The text for completed operations which have been applied to the requirement relative to the DBMS Protection Profile [DPP] or relative to Part 2 of [CC] (for SFRs that are not in [DPP]) is highlighted with *ITALICISED CAPITAL LETTERS* within each requirement. Annex B provides definitions for various terms used in the functional requirements.

Table 3: List of Security Functional Requirements

Component	Name	A	S	R	I
FAU_GEN.1	Audit Data Generation	X		X	
FAU_GEN.2	User Identity Association				
FAU_SAR.1	Audit Review				
FAU_SAR.3	Selectable Audit Review	X			
FAU_SEL.1	Selective Audit	X			
FAU_STG.1	Protected Audit Trail Storage				
FAU_STG.4	Prevention of Audit Data Loss	X		X	

Component	Name	A	S	R	I
FDP_ACC.1	Subset Access Control				
FDP_ACF.1	Security Attribute Based Access Control	X			
FDP_RIP.2	Subset Residual Information Protection			X	
FIA_AFL.1	Basic Authentication Failure Handling	X			
FIA_ATD.1	User Attribute Definition	X			
FIA_SOS.1	Verification of Secrets	X			
FIA_UAU.1	Timing of Authentication	X			
FIA_UID.1	Timing of Identification	X			
FIA_USB.1	User-Subject Binding				
FMT_MSA.1	Management of Security Attributes	X		X	X
FMT_MSA.3	Static Attribute Initialisation	X		X	X
FMT_MTD.1	Management of TSF Data			X	
FMT_REV.1	Revocation	X			
FMT_SMR.1	Security Roles	X			
FPT_RVM.1	Non-Bypassability of the TSP				
FPT_SEP.1	TSF Domain Separation				
FRU_RSA.1	Maximum Quotas	X			
FTA_MCS.1	Basic Limitation on Multiple Concurrent Sessions	X			
FTA_TSE.1	TOE Session Establishment	X			
FDP_IFC.1 *	Subset Information Flow Control	X			
FDP_IFF.2 *	Hierarchical Security Attributes	X		X	
FMT_MOF.1 *	Management of Security Functions Behaviour	X	X		

Note that FMT_MSA.1.1.2, FMT_MSA.3.1.2 and FMT_MSA.3.2.2 are SFR elements that are not included in [DPP] and have been added to cover requirements for the management of security attributes associated with Label-Based Access Control. They are defined in the Section “SFRs Additional to those in [DPP]” towards the end of this Chapter. SFR elements FMT_MSA.1.1.1, FMT_MSA.3.1.1 and FMT_MSA.3.2.1 specify identical requirements to SFRs FMT_MSA.1.1, FMT_MSA.3.1 and FMT_MSA.3.2 that are in [DPP].

Note also that there is the possibility of confusion between the Common Criteria [CC] term “policy” and the OLS term “policy”. The Common Criteria term is used in the context of the phrase “Security Function Policy” (SFP) which is the security policy enforced by a particular Security Function (SF). OLS policies are established by a database administrator to specify how Label-Based Access Control is to be enforced

on a database. Such a policy will always be referred to in this document via the phrase "OLS policy".

Security Audit

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

a) Start-up and shutdown of the database audit functions;
b) All auditable events for the basic level of audit as identified in Tables 4 and 7 of [DPP] *AND TABLE 4 BELOW*; and
c) *NO ADDITIONAL EVENTS*.

Table 4: List of LBAC Functions' Auditable Events

Component	Event	Additional Data
FDP_IFC.1	None (i.e. no such events are to be audited)	None
FDP_IFF.2	All decisions on requests for information flow	None
FMT_MOF.1	All modifications in the behaviour of the functions in the TSF	None
FMT_MSA.1	All modifications of the values of *DATABASE OBJECT LABELS*	*NEW DATABASE OBJECT LABEL*
FMT_MSA.3	Modifications of the default setting of permissive or restrictive *DATABASE OBJECT LABEL* rules	None

FAU_GEN.1.2 The TSF shall record within each database audit record at least the following information:

a) Date and time of the database event, type of database event, database subject identity, and the outcome (success or failure) of the event;
b) The current session label for each OLS policy defined for the database; and
c) For each database audit event type, based on the auditable event definitions of the functional components included in the PP/ST, *OTHER RELEVANT INFORMATION AS IDENTIFIED IN TABLES 4 & 7 OF [DPP] AND TABLE 4 ABOVE*.

FAU_GEN.2.1 The TSF shall be able to associate each auditable database event with the identity of the database user that caused the event.

FAU_SAR.1.1 The TSF shall provide authorised database users with the capability to read all database audit information from the database audit records.

FAU_SAR.1.2 The TSF shall provide the database audit records in a manner suitable for the database user to interpret the information.

FAU_SAR.3.1 The TSF shall provide the ability to perform searches and sorting of database audit data based on *THE VALUES OF AUDIT DATA FIELDS*.

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

a) event type;
b) database subject identity;
c) database object identity;
d) *DATABASE SYSTEM PRIVILEGE*.

FAU_STG.1.1 The TSF shall protect the stored database audit records from unauthorised deletion.

FAU_STG.1.2 The TSF shall be able to prevent modifications to the database audit records.

FAU_STG.4.1 The TSF shall prevent *DATABASE AUDIT* events, except those taken by the authorised *DATABASE* user with special rights, *IF THE AUDIT TRAIL IS FULL*.

User Data Protection

FDP_ACC.1.1 The TSF shall enforce the database object access control SFP on:

a) database subjects;
b) database objects;
c) all permitted operations on database objects by database subjects covered by the SFP.

Note that the Label-Based Access Control SFP is also to be applied to database subjects, objects and operations as specified in SFR FDP_IFC.1.1 and SFRs FDP_IFF.2.1 to FDP_IFF.2.7. These SFRs are given in the "SFRs Additional to those in [DPP]" section near the end of this chapter. The Label-Based Access Control SFP applies controls that are additional to the database object access control SFP.

FDP_ACF.1.1 The TSF shall enforce the database object access control SFP to database objects based on:

a) the identity of the owner of the database object; and
b) the object access privileges to the database object held by the database subject; and
c) the database administrative privileges of the database subject.

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled database subjects and controlled database objects is allowed:

a) if the user associated with the database subject is the owner of the database object, then the requested access is allowed; or
b) if the database subject has the database object access privilege for the requested access to the database object, then the requested access is allowed; or
c) otherwise access is denied, unless access is explicitly authorised in accordance with the rules specified in FDP_ACF.1.3.

FDP_ACF.1.3 The TSF shall explicitly authorise access of database subjects to database objects based on the following additional rules:

a) if the database subject has a database administrative privilege to override the database object access controls for the requested access to the database object, then the requested access is allowed;
b) *IF THE SUBJECT IS CONNECTED AS SYSBDA THEN THE REQUESTED ACCESS IS ALLOWED; OR*

c) *IF THE SUBJECT IS CONNECTED AS SYSOPER AND THE REQUESTED ACTION IS ONE OF THE OPERATIONS PERMITTED FOR THE SYSOPER USER SPECIFIED IN [DAG, 1-13], THEN THE REQUESTED ACCESS IS ALLOWED.*

FDP_ACF.1.4 The TSF shall explicitly deny access of database subjects to database objects based on the following additional rules: NONE.

FDP_RIP.2.1 The TSF shall ensure that any previous information content of a database resource is made unavailable upon the allocation of a resource to *SCHEMA OBJECTS (INCLUDING NON-SCHEMA OBJECTS, WHICH ARE STORED IN THE SYS SCHEMA)*.

Identification and Authentication

FIA_AFL.1.1 The TSF shall detect when *A NUMBER, CONFIGURED BY AN AUTHORISED ADMINISTRATIVE USER*, of unsuccessful database authentication attempts occur related to *THE USER'S LAST SUCCESSFUL DATABASE SESSION*.

FIA_AFL.1.2 When the defined number of unsuccessful database authentication attempts has been met or surpassed, the TSF shall *LOCK THE DATABASE USER'S ACCOUNT*.

FIA_ATD.1.1 The TSF shall maintain the following list of security attributes belonging to individual database users:

a) database user identity;
b) database object access privileges;
c) database administrative privileges;
d) *ORACLE ROLES;*
e) *AND FOR EACH OLS POLICY FOR WHICH THE USER HAS AUTHORISATION:*
A MAXIMUM LEVEL;
A MINIMUM LEVEL;
A (POSSIBLY EMPTY) SET OF AUTHORISED COMPARTMENTS;
FOR EACH AUTHORISED COMPARTMENT, A SPECIFICATION OF READ ACCESS OR READ-WRITE ACCESS;
A (POSSIBLY EMPTY) SET OF AUTHORISED GROUPS;
FOR EACH AUTHORISED GROUP, A SPECIFICATION OF READ ACCESS OR READ-WRITE ACCESS;
AN INITIAL SESSION LABEL;
A (POSSIBLY EMPTY) SET OF LABEL-BASED ACCESS CONTROL PRIVILEGES.

FIA_SOS.1.1 The TSF shall provide a mechanism to verify that database secrets (passwords) meet *REUSE, LIFETIME, AND CONTENT METRICS AS DEFINED BY AN AUTHORISED ADMINISTRATIVE USER*.

FIA_UAU.1.1 The TSF shall allow *THE FOLLOWING LIST OF ACTIONS* on behalf of the database user to be performed before the database user is authenticated:

a) OBTAIN THE CURRENT ORACLE VERSION STRING AND NUMBER;
b) ESTABLISH A DATABASE CONNECTION; AND
c) RECEIVE AN ERROR MESSAGE UPON ERROR.

FIA_UAU.1.2 The TSF shall require each database user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that database user.

FIA_UID.1.1 The TSF shall allow *THE FOLLOWING LIST OF ACTIONS* on behalf of the database user before the database user is identified:

a) OBTAIN THE CURRENT ORACLE VERSION STRING AND NUMBER;
b) ESTABLISH A DATABASE CONNECTION; AND
c) RECEIVE ERROR MESSAGES UPON ERROR.

FIA_UID.1.2 The TSF shall require each database user to be successfully identified before allowing any other TSF-mediated actions on behalf of that database user.

FIA_USB.1.1 The TSF shall associate the appropriate database user security attributes with database subjects acting on behalf of that database user.

Security Management

FMT_MSA.1.1.1 The TSF shall enforce the database object access control SFP to restrict the ability to modify the database object security attributes:

a) *DATABASE OBJECT ACCESS PRIVILEGES* to *THE OBJECT'S OWNER AND OTHER DATABASE USERS AUTHORIZED BY THE OWNER*.
b) *DATABASE SYSTEM PRIVILEGES* to *USERS WHO HAVE BEEN GRANTED THAT PRIVILEGE WITH ADMIN OPTION OR USERS WHO CONNECT AS SYSDBA*.
c) *DATABASE ROLES* to *DATABASE USERS AUTHORIZED TO MODIFY ROLES*.

FMT_MSA.3.1.1 The TSF shall enforce the database object access control SFP to provide restrictive default values for database object security attributes that are used to enforce the database object access control SFP.

FMT_MSA.3.2.1 The TSF shall allow *NO DATABASE USERS* to specify alternative initial values to override the default values when a database object is created.

FMT_MTD.1.1 The TSF shall, according to Tables 5 and 8 of [DPP] *AND TABLE 5 BELOW,* restrict the ability to perform operations on TSF data to database administrative users.

Table 5: List of LBAC Functions' Required Management Events

Component	Operation	TSF Data
FDP_IFC.1	-	-
FDP_IFF.2	Managing	The attributes used to make explicit access or denial based decisions
FMT_MOF.1	Managing	The group of roles that can interact with the functions in the TSF
FMT_MSA.1	Manage	The group of database roles that can interact with the *DATABASE OBJECT LABELS*

Table 5: List of LBAC Functions' Required Management Events

Component	Operation	TSF Data
FMT_MSA.3	Manage	The permissive or restrictive setting of default values for the *LABEL-BASED ACCESS CONTROL* SFP

FMT_REV.1.1 The TSF shall restrict the ability to revoke security attributes associated with the database users and database objects within the TSC to:

a) authorised database administrators (for users and objects);
b) authorised database users (only for the database objects they own or database objects for which they have been granted database object access privileges allowing them to revoke security attributes);
c) *NO OTHER ROLES.*

FMT_REV.1.2 The TSF shall enforce the *FOLLOWING* rules:

a) revocation of database object access privileges shall take effect prior to all subsequent attempts to establish access to the database object;
b) revocation of database administrative privileges shall take effect prior to when the user begins the next database session;
c) *NO ADDITIONAL REVOCATION RULES.*

FMT_SMR.1.1 The TSF shall maintain the database roles:

a) database administrative user;
b) database user;
c) *DATABASE ROLES DEFINED BY SUITABLY PRIVILEGED DATABASE ADMINISTRATIVE USERS.*

Note that due to a difference in terminology between the CC and the Oracle9i product the two occurrences of the word "role" in FMT_SMR.1.1 have different meanings. The first occurrence, which is part of the required CC wording, is a general term meaning any kind of user that can be created within the TSF. The second occurrence, which is part of a completed assignment in [DPP], is a specific term referring to Oracle9i database roles that can be configured and granted to users of the Oracle9i product.

FMT_SMR.1.2 The TSF shall be able to associate database users with database roles.

Protection of the TOE Security Functions

FPT_RVM.1.1 The TSF shall ensure that TSP enforcement functions are invoked and succeed before each function within the TSC is allowed to proceed.

FPT_SEP.1.1 The TSF shall maintain a security domain for its own execution that protects it from interference and tampering by untrusted database subjects.

FPT_SEP.1.2 The TSF shall enforce separation between the security domains of database subjects in the TSC.

Resource Utilisation

FRU_RSA.1.1 The TSF shall enforce maximum quotas of the following resources: *a) CPU_TIME;*

b) ELAPSED TIME;
c) LOGICAL DATA BLOCKS READ; AND
d) DATABASE STORAGE ALLOCATED.

that an individual database user can use over a specified period of time.

TOE Access

FTA_MCS.1.1 The TSF shall restrict the maximum number of concurrent database sessions that belong to the same database user.

FTA_MCS.1.2 The TSF shall enforce, by default, a limit of *A NUMBER, CONFIGURED BY AN AUTHORIZED ADMINISTRATIVE USER*, database sessions per database user.

FTA_TSE.1.1 The TSF shall be able to deny database session establishment based on *USER IDENTITY*.

Note that the DBA and OPER users can always connect to the database.

SFRs Additional to those in [DPP]:

.

User Data Protection

FDP_IFC.1.1 The TSF shall enforce the *LABEL-BASED ACCESS CONTROL SFP* on:

a) DATABASE SUBJECTS;
b) LABELLED DATABASE OBJECTS;
c) ALL PERMITTED OPERATIONS ON LABELLED OBJECTS BY A DATABASE SUBJECT COVERED BY THE SFP.

FDP_IFF.2.1 The TSF shall enforce the *LABEL-BASED ACCESS CONTROL SFP* based on the following types of subject and information security attributes:

a) DATABASE SUBJECT LABELS; AND
b) LABELS OF THE DATABASE OBJECT CONTAINING THE INFORMATION.

Note: Labels shall include an hierarchic classification level and a (possibly empty) set of non-hierarchic categories and a (possibly empty) set of hierarchic groups. An object is to have one label for each OLS policy that applies to it.

FDP_IFF.2.2 The TSF shall permit an information flow between a controlled subject and a controlled object via a controlled operation if the following rules, based on the ordering relationships between security attributes, hold:

a) A DATABASE SUBJECT MAY OBSERVE THE CONTENTS OF A DATABASE OBJECT ONLY IF, FOR EVERY OLS POLICY THAT APPLIES TO THE OBJECT:
READ_CONTROL FOR THE POLICY IS OFF OR THE SESSION LABEL OF THE DATABASE SUBJECT DOMINATES THE LABEL OF THE DATABASE OBJECT; AND
b) A DATABASE SUBJECT MAY MODIFY A DATABASE OBJECT ONLY IF, FOR EVERY OLS POLICY THAT APPLIES TO THE OBJECT:
THE RELEVANT WRITE_CONTROL IS OFF FOR THE

POLICY
OR
IF THE POLICY WAS NOT CREATED WITH THE INVERSE GROUP OPTION,
THEN
(*THE LEVEL IN THE OBJECT'S LABEL IS GREATER THAN OR EQUAL TO THE SUBJECT'S MINIMUM LEVEL AND LESS THAN OR EQUAL TO THE SUBJECT'S SESSION LEVEL,*
AND
(THE OBJECT'S LABEL CONTAINS GROUPS AND THE SUBJECT'S LABEL ALLOWS WRITE ACCESS TO ONE OF THE GROUPS (OR ITS PARENT) IN THE OBJECT'S LABEL AND THE SUBJECT'S LABEL INCLUDES ALL THE COMPARTMENTS IN THE OBJECT'S LABEL,
OR
THE OBJECT'S LABEL CONTAINS NO GROUPS AND THE SUBJECT'S LABEL ALLOWS WRITE ACCESS TO ALL THE COMPARTMENTS IN THE OBJECT'S LABEL
))
ELSE
(*THE LEVEL IN THE OBJECT'S LABEL IS GREATER THAN OR EQUAL TO THE SUBJECT'S MINIMUM LEVEL AND LESS THAN OR EQUAL TO THE SUBJECT'S SESSION LEVEL,*
AND
THE GROUPS IN THE OBJECT'S LABEL FORM A SUPERSET OF THE GROUPS IN THE SUBJECT'S LABEL,
AND
THE MAXIMUM SET OF AUTHORISED INVERSE GROUPS THAT CAN BE SET IN ANY SUBJECT'S SESSION LABEL IS A SUPERSET OF THE GROUPS IN THE OBJECT'S LABEL
AND
THE SUBJECT'S LABEL ALLOWS WRITE ACCESS TO ALL THE COMPARTMENTS IN THE OBJECT'S LABEL
).

Note: OLS policies assigned to objects shall specify which controls are to be applied when a subject attempts to access an object.
Note also that the phrase "OR ITS PARENT" in the above SFR is to be taken to mean "OR ITS PARENT OR ITS PARENT'S PARENT OR ITS PARENT'S PARENT'S PARENT ETC.".

FDP_IFF.2.3 The TSF shall *ALLOW A USER TO CHANGE THE SESSION LABEL TO A COMBINATION OF ANY OF THE USER'S AUTHORISED COMPARTMENTS AND GROUPS WITH A LEVEL IN THE RANGE BOUNDED BY THE USER'S MAXIMUM AND MINIMUM LEVEL*.

FDP_IFF.2.4 The TSF shall provide the following *ADDITIONAL SFP CAPABILITY*:
THE TSF WILL EXECUTE A STORED PROCEDURE, FUNCTION OR PACKAGE AT THE EXECUTING USER'S CURRENT SESSION LABEL AND WITH THE SET OF LABEL-BASED

ACCESS CONTROL PRIVILEGES FORMED BY THE UNION OF THE PRIVILEGES OF THE EXECUTING USER AND THE PRIVILEGES GIVEN TO THE STORED PROCEDURE, FUNCTION OR PACKAGE.

FDP_IFF.2.5 The TSF shall explicitly authorise an information flow based on the following rule:
IF THE SUBJECT HAS THE APPROPRIATE LABEL-BASED ACCESS CONTROL PRIVILEGE FOR THE OPERATION, THEN THE INFORMATION FLOW WILL BE PERMITTED.

FDP_IFF.2.6 The TSF shall *ENFORCE NO ADDITIONAL RULES TO* explicitly deny an information flow.

FDP_IFF.2.7 The TSF shall provide the following relationships for any two valid *LABELS*:

a) There exists an ordering function that, given two valid *LABELS*, determines if the *LABELS* are equal, if one *LABEL* is greater than the other, or if the *LABELS* are incomparable; and.

b) There exists a "least upper bound" in the set of *LABELS*, such that, given any two valid *LABELS*, there is a valid *LABEL* that is greater than or equal to the two valid *LABELS*; and

c) There exists a "greatest lower bound" in the set of *LABELS*, such that, given any two valid *LABELS*, there is a valid *LABEL* that is not greater than the two valid *LABELS.*

Note: The TSF is to supply an ordering function "greater than" whereby Label1 is greater than Label2 if Label1 dominates Label2 and Label1 is not equal to Label2. Label1 and Label2 are incomparable if Label1 does not dominate Label2 and Label2 does not dominate Label1.

Security Management

FMT_MOF.1.1 The TSF shall restrict the ability to *MODIFY THE BEHAVIOUR OF* the *LABEL-BASED ACCESS CONTROL* functions to *AUTHORISED ADMINISTRATIVE USERS.*

FMT_MSA.1.1.2 The TSF shall enforce the *LABEL-BASED ACCESS CONTROL SFP* to restrict the ability to *MODIFY LABELS AND PRIVILEGES* to *SUITABLY PRIVILEGED USERS.*

FMT_MSA.3.1.2 The TSF shall enforce the *LABEL-BASED ACCESS CONTROL SFP* to provide *NO* default values for *DATABASE OBJECT* security attributes that are used to enforce the *LABEL-BASED ACCESS CONTROL SFP.*

Note: The TSF is to ensure that, when a user creates an object which is controlled by the Label-Based Access Control SFP, a value must be specified for the label.

FMT_MSA.3.2.2 The TSF shall allow *NO DATABASE USERS* to specify alternative initial values to override the default values *FOR LABEL-BASED ACCESS CONTROL SECURITY ATTRIBUTES* when a database object is created.

Note: The TSF is to ensure that, when an object is created which is controlled by the Label-Based Access Control SFP, no database user can cause a value to be given to the label other than that specified for the label in conformance with the rules of the SFP.

TOE Security Assurance Requirements

The target assurance level is EAL4 as defined in Part 3 of the CC, augmented with ALC_FLR.3 as defined in [CEM_FLR].

Security Requirements for the IT Environment

As per [DPP 5.5 & 5.6], except that [DPP, 5.6] only applies for Microsoft Windows platforms (because it relates to OS Authentication requirements and the OS Authentication package is only claimed for Microsoft Windows).

Minimum Strength of Function

The minimum strength of function for the TOE is *SOF-High*. This exceeds the requirements in [DPP].

This Page Intentionally Blank

CHAPTER

6 TOE Summary Specification

TOE Security Functionality

This section contains a high-level specification of each Security Function (SF) of the TOE that contributes to satisfaction of the Security Functional Requirements of chapter 5. The specifications cover five major areas: identification and authentication, database resource quotas, access controls, privileges and roles, and auditing.

Table 6 below shows that all the SFRs are satisfied by at least one SF and that every SF is used to satisfy at least one SFR (but note that SFRs FDP_ACF.1.4 and FDP_IFF.2.6 are not satisfied by any particular SF because these SFRs specify null functionality).

Table 6: Mapping of SFs to SFRs in [DPP]

	FIA									FDP						FMT								FPT			FRU	FTA			FAU									
	AFL.1.1	AFL.1.2	ATD.1.1	SOS.1.1	UAU.1.1	UAU.1.2	UID.1.1	UID.1.2	USB.1.1	ACC.1.1	ACE.1.1	ACE.1.2	ACE.1.3	ACE.1.4	RIP.2.1	MSA.1.1.1	MSA.3.1.1	MSA.3.2.1	MTD.1.1	REV.1.1	REV.1.2	SMR.1.1	SMR.1.2	RVM.1.1	SEP.1.1	SEP.1.2	RSA.1.1	MCS.1.1	MCS.1.2	TSE.1.1	GEN.1.1	GEN.1.2	GEN.2.1	SAR.1.1	SAR.1.2	SAR.3.1	SEL.1.1	STG.1.1	STG.1.2	STG.4.1
F.IA.PRE					Y	Y	Y	Y																															
F.IA.UID								Y														Y																	
F.IA.DBA						Y		Y																															
F.IA.OSA						Y		Y																															
F.IA.CNF												Y																											
F.IA.IDE								Y																Y	Y	Y													
F.IA.CSA						Y		Y														Y	Y							Y									
F.IA.CSN						Y		Y														Y								Y									
F.IA.PWD	Y	Y		Y																																			
F.IA.ATT			Y						Y										Y																				
F.IA.USE				Y																																			
F.IA.POLICY			Y						Y										Y																				
F.IA.SESSION									Y																														
F.IA.SESSUPD									Y																														
F.LIM.CNF				Y															Y	Y							Y												
F.LIM.POL																											Y		Y										
F.LIM.NSESS																											Y	Y	Y										
F.LIM.TIME																											Y												
F.LIM.RSESS																											Y												
F.LIM.RCALL																											Y												
F.ACCESS										Y														Y															
F.DAC.OBID										Y	Y	Y																											
F.DAC.OBREF										Y	Y	Y																											
F.DAC.SUA										Y	Y																												
F.DAC.OBA										Y	Y							Y																					
F.DAC.POL											Y	Y	Y				Y	Y						Y															
F.DAC.SEP																									Y	Y													
F.DAC.OR															Y																								
F.LBAC.POL																								Y															
F.LBAC.LABSET																			Y																				
F.LBAC.LABUPD									Y										Y																				
F.LBAC.REF																																							
F.LBAC.TRIGGER									Y																														
F.LBAC.XVP									Y																														
F.LBAC.MOD																																							
F.APR.GOP																Y		Y	Y																				
F.APR.ROP																Y			Y	Y																			
F.APR.GRSP																Y		Y	Y	Y																			
F.APR.GRPP																		Y	Y	Y																			
F.APR.GRR																Y		Y	Y	Y		Y																	
F.APR.DER																							Y																
F.APR.EDR									Y														Y																
F.PRI.SPRIV									Y		Y	Y	Y				Y				Y																		
F.PRI.PPRIV									Y																														
F.PRI.XVP									Y																														
F.PRI.PRX									Y														Y																
F.AUD.SOM																														Y					Y				
F.AUD.SEV																														Y					Y				
F.AUD.ALW																														Y					Y				
F.AUD.CNF												Y	Y																						Y				
F.AUD.ACC																			Y														Y		Y				
F.AUD.DEL																			Y																	Y	Y		
F.AUD.INF																															Y	Y							
F.AUD.LCOL																															Y								
F.AUD.LAUD																														Y									
F.AUD.LEN																														Y									
F.AUD.LDIS																														Y									
F.AUD.VIEW																																		Y	Y				
F.AUD.LVIEW																																		Y	Y				
F.AUD.FULL																																							Y

Table 7: Mapping of SFs to SFRs Additional to those in [DPP]

	FDP								FMT			
	IFC.1.1	IFF.2.1	IFF.2.2	IFF.2.3	IFF.2.4	IFF.2.5	IFF.2.6	IFF.2.7	MOF.1.1	MSA.1.1.2	MSA.3.1.2	MSA.3.2.2
F.IA.PRE												
F.IA.UID												
F.IA.DBA												
F.IA.OSA												
F.IA.CNF												
F.IA.IDE												
F.IA.CSA												
F.IA.CSN												
F.IA.PWD												
F.IA.ATT												
F.IA.USE												
F.IA.POLICY										Y		
F.IA.SESSION												
F.IA.SESSUPD				Y								
F.LIM.CNF												
F.LIM.POL												
F.LIM.NSESS												
F.LIM.TIME												
F.LIM.RSESS												
F.LIM.RCALL												
F.ACCESS	Y											
F.DAC.OBID												
F.DAC.OBREF												
F.DAC.SUA												
F.DAC.OBA												
F.DAC.POL												
F.DAC.SEP												
F.DAC.OR												
F.LBAC.POL	Y	Y	Y			Y		Y				
F.LBAC.LABSET											Y	Y
F.LBAC.LABUPD										Y		
F.LBAC.REF	Y											
F.LBAC.TRIGGER												
F.LBAC.XVP					Y							
F.LBAC.MOD									Y			
F.APR.GOP												
F.APR.ROP												
F.APR.GRSP												
F.APR.GRPP										Y		
F.APR.GRR												
F.APR.DER												
F.APR.EDR												
F.PRI.SPRIV												
F.PRI.PPRIV						Y						
F.PRI.XVP												
F.PRI.PRX												
F.AUD.SOM												
F.AUD.SEV												
F.AUD.ALW												
F.AUD.CNF												
F.AUD.ACC												
F.AUD.DEL												
F.AUD.INF												
F.AUD.LCOL												
F.AUD.LAUD												
F.AUD.LEN												
F.AUD.LDIS												
F.AUD.VIEW												
F.AUD.LVIEW												
F.AUD.FULL												

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 and authenticating the user.

Note that users can obtain the current Oracle version string and version number by calling OCIServerVersion, as described in [OCI, 16: OCIServerVersion].

F.IA.UID Each database user is uniquely identified.

F.IA.DBA DBMS Identification and Authentication:

If a user is configured in the TOE as being *identified by* a password then the TOE will:

a) identify the user by confirming that the user provides a valid user identifier, and

b) authenticate the user by confirming that the user provides a password corresponding to the stored password for that user.

F.IA.OSA OS Identification and Authentication:

If a user is configured in the TOE as being *identified externally* then the TOE will identify and authenticate the user by confirming that the requesting subject's OS user identifier, prefixed by the value of the OS_AUTHENT_PREFIX initialisation parameter, matches a database user identifier.

Note that in F.IA.OSA the TOE is not performing any authentication, per se. Rather the TOE is dependent on the OS to correctly authenticate the user.

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 and the TOE has identified and authenticated the user as a valid database user (by either DBMS or OS identification and authentication).

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 a connection or connected to an instance during the course of a database session.

F.IA.CSA The TOE will create a database session as the SYS user (for AS SYSDBA connections) or the PUBLIC user (for AS SYSOPER connections) only if either:

a) the requesting subject has the platform-specific access rights for OSDBA and OSOPER, respectively, as defined in

[DAG, 1: Database Administrator Authentication (OSDBA and OSOPER)], or

b) the provided user identifier and password correspond to users stored in the Oracle password file as being allowed SYSDBA or SYSOPER connections, respectively.

F.IA.CNF The TOE will allow only a suitably authorised user to create a database user.

F.IA.PWD The TOE provides the following configurable controls on user passwords: [SQL, 14: CREATE PROFILE]

a) the number of failed login attempts before the user account is locked,

b) the number of days the same password can be used before expiring,

c) the number of days before which a password cannot be reused,

d) the number of password changes required before the current password can be reused,

e) the number of days a user account will be locked after the specified number of consecutive failed logins,

f) the number of days of grace period after a password expires before the user account is locked,

g) a password complexity check to screen passwords selected by the user.

F.IA.ATT The data dictionary contains a unique set of security attributes for each user including their username, password management information, account status (i.e. locked or unlocked), privileges, roles and resource limits that can be displayed and modified by suitably authorised users using standard SQL commands.

F.IA.USE A database user is authorised to change the password associated with that user within the following constraints:

a) If the user's profile includes a complexity check function, then the new password is accepted only if it meets the criteria of the complexity check.

b) If the user's profile specifies password reuse constraints and the user attempts to reuse a password, the TOE rejects the change if the reuse constraints are not met. [SQL, 14: CREATE PROFILE].

F.IA.POLICY For each OLS policy defined for the database, the data dictionary contains a set of security attributes for each user authorised to access data protected by that policy. These security attributes include the user label authorisations, initial session label, initial default row label and policy privileges. The user label authorisations consist of a maximum and minimum level, a set of authorised compartments, a set of authorised groups, and, for each such compartment and group, a specification of read-only

access or read-write access. When first created, a user has no such security attributes, but they can be set and modified by suitably authorised users.

F.IA.SESSION When starting a new database session, the session label and default row label for each applicable OLS policy are set as follows:

a) When a user connects to the database, for each OLS policy for which the user is authorised, the TOE will set the session label and default row label for the user's session using the user's initial session label and initial default row label attributes defined for the policy in the data dictionary;

b) If a user is already connected to the database, but uses OCI to begin a new database session, for each OLS policy for which there is a SYS_CONTEXT, if the SYS_CONTEXT variables INITIAL_LABEL and INITIAL_ROW_LABEL are within the user's label authorisations, then they are used instead of the user's attributes in the data dictionary in setting the session label and default row label.

F.IA.SESSUPD A user can change the session label and default row label for the user's session provided these labels remain within the user's label authorisations.

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.

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.ACCESS For all attempts by subjects to access objects which are subject to the administration of rights, the TOE shall:

a) verify the validity of the request on the basis of the discretionary access control policy and, if the object has a label, the label-based access control policy; and

b) reject the attempt if either the discretionary or the label-based access checks fail.

Note that if the discretionary access check fails, the label-based access check will not be made.

Discretionary Access Control

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

F.DAC.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.DAC.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;

c) and any security policies providing fine-grained access control for the object.

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

---

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.

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 [SQL, 16: GRANT (*grant_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 [SQL, 16: GRANT (*grant_system_privileges)*] and provide the ability to restrict a user's use of operations to those operations which do not modify objects.

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

e) If the user is connected AS SYSOPER and the operation is one of those specified in [DAG, 1: Database Administrator Authentication (OSDBA and OSOPER)], for the OSOPER role then the requested access is allowed.

F.DAC.SEP The TOE does not allow interference between concurrent database sessions.

F.DAC.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 [SQL, 2].

Label-Based Access Control

F.LBAC.POL The label-based access policy of the TOE shall permit a subject to access an object which has a label only if, for all OLS policies protecting the object:

a) the LBAC access mediation rules permit the subject to perform the operation as follows:

observation of the contents of a database object by a database subject is governed by the rules as specified in FDP_IFF.2.2a, and elaborated in [OLSAG, 3-13 to 3-14 and 7-6 and 13-9 to 13-10],

modification of a database object by a database subject is governed by the rules as specified in FDP_IFF.2.2b, and

elaborated in [OLSAG, 3-15 to 3-17 and 7-6 and 13-11 to 13-12]; or

b) the subject's database session has the necessary OLS policy privileges which enable override of the LBAC access mediation rules (see [OLSAG, 3-18 to 3-22 and 13-13 to 13-15]); or

c) the user is SYS or LBACSYS or is connected AS SYSDBA; or

d) the subject's database session has the system privilege EXEMPT ACCESS POLICY effective.

Note that the LBAC policy applies to subjects which are database users and processes and tasks running on behalf of such users and applies to objects which are rows in tables that have been assigned one or more OLS policies. Further details on the LBAC policy are provided in:

- *[OLSAG, 7-2 to 7-7 and 13-4], which describes the various policy options*
- *[OLSAG, 7-9], which describes the exemptions that are allowed from OLS policy enforcement*
- *[OLSAG, 7-10 to 7-17 and 13], which describes how the enforcement options and labelling functions affect the insertion, update and deletion of labelled data*
- *[OLSAG, 7-18 to 7-19], which describes the use of SQL predicates with an LBAC policy*
- *[OLSAG, 4-12 to 4-13 and 13-26], which describes the Least Upper Bound and Greatest Lower Bound functions which relate to the dominance relationship used for some of the LBAC mediation rules*
- *[OLSAG, A-2 to A-6 and 13-27], which describes functions to calculate whether a label dominates another label*
- *[OLSAG, 13] which describes the releasabilities scheme which is implemented via the INVERSE_GROUP policy enforcement option .*

Note that an implication of this SF is that a subject can only access an object that has been put under the protection of more than one OLS policy if the LBAC mediation rules for all of these OLS policies permit the subject to access the object.

F.LBAC.LABSET When inserting a row in a table protected by an OLS policy, the row's label for each such policy is set according to the enforcement options defined for the policy (see [OLSAG, 4-16 to 4-18 and 7-5 to 7-11]).

F.LBAC.LABUPD Attempts to update the label of a row in a table protected by an OLS policy are subject to the enforcement options defined for the policy (see [OLSAG, 7-3, 7-5 to 7-6 and 7-14 to 7-16]).

F.LBAC.REF If a child row is being inserted or updated when the parent row is in a table protected by an OLS policy, then if the child row is in a table which has a referential integrity constraint, the user must have LBAC read access to the parent row.

F.LBAC.TRIGGER The TOE will execute a trigger with the session label and with the policy privileges of the user that invoked the trigger.

F.LBAC.XVP The TOE will execute a stored procedure, function or package with the user's session label and with the set of OLS policy privileges which is the union of:

a) the OLS policy privileges of the executing user; and

b) the OLS policy privileges assigned to the stored procedure, function or package.

Note that if another stored procedure, function or package (which is known as a "stored program unit") is called within the execution of the original stored program unit, it runs with the same OLS policy privileges as the original stored program unit.

F.LBAC.MOD The TOE only allows suitably privileged users to modify or delete the packages that implement LBAC.

Note that only trusted administrators have sufficient privilege to affect the way LBAC operates by modifying or deleting the relevant packages.

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 connected AS SYSDBA; *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.GRPP For a given OLS policy, *policy*, a user (the grantor) can grant a policy privilege to another user or to a stored program unit and can revoke a policy privilege from the grantee, only if the grantor (or revoker) has been granted the *policy*_DBA role and has the EXECUTE privilege for the SA_USER_ADMIN package.

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 connected AS SYSDBA; *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.

Note that c) includes the case where the grantor is the user who created the role - see [DAG, 25: Granting System Privileges and Roles (Granting the ADMIN OPTION)]: "When a user creates a role, the role is automatically granted to the creator with the ADMIN OPTION." and the warning on [DAG, 25: Specifying Default Roles] which adds the fact that a role is automatically granted to its creator as a default role.

Enabling and Disabling Roles

F.APR.DER A role can be granted to a user in one of the following ways:

a) As a default role, in which case the role will be enabled automatically for each database session created by that user[1].

b) As a non-default role, in which case

i. if the role is configured in the TOE as being *identified* using a package, then that package 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; or

ii. if the role is configured in the TOE as being *not identified,* then 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.

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[2]), 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.

Effective Privileges

F.PRI.SPRIV An object or system privilege will be effective in a user session only if:

a) the privilege was granted to the user directly and has not been revoked from the user; *or*

b) the privilege was granted indirectly via the PUBLIC user group and has not been revoked from PUBLIC; *or*

1. A default role is enabled at session creation bypassing any authorisation required for that role.

2. When a role that has been granted other roles is enabled all the indirectly granted roles are implicitly enabled.

c) the privilege was granted to the user indirectly via a role, and has not been revoked from that role and the role is effective in the current session.

F.PRI.PPRIV An OLS policy privilege will be effective for the policy in a user session only if the privilege was set in the user's data dictionary entry for the policy before the start of the session.

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 directly granted privileges of the owning user) to allow other users to have controlled access to specified aggregations of data.

F.PRI.PRX A suitably authorised user can provide other users with the ability to establish a proxy connection for another user. The authorised user can control which user roles are available to the proxy session.

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;

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;

d) by use of data access based on content;

e) 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;

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

g) for each event of type a) and c);

i. for all users;

ii. for specified users;

iii. for specified proxies on behalf of any user;

iv. for specified proxies on behalf of specified users;

F.AUD.ALW Irrespective of the TOE's audit configuration, the TOE will audit every successful occurrence of the following events to the operating system:

a) start-up;

b) shut-down;

c) connection through the keywords AS SYSDBA or AS SYSOPER.

Note that, for the Solaris platform, OS auditable Oracle records are written to standard text file audit logs in the OS. For the Windows NT platform, OS auditable Oracle records are written to the Windows NT event log.

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

Note that, by default (after installation), the TOE allows only SYS and SYSTEM (who are granted the DBA role during installation) and users connected AS SYSDBA *to set and alter the audit configuration. It is possible for these users to grant the relevant privileges to other users, but it is assumed that they will not do this in practice.*

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

a) any suitably authorised 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[1] user to delete or update 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 object accessed; operation performed or attempted; outcome of the operation; system privileges used.

In particular:

a) when a user attempts a connection to a database, whether successful or not, at least the following information is recorded when the TOE is configured to audit connection

1. By default, the TOE allows only SYS and SYSTEM (who are granted the DBA role during installation) and users connected AS SYSDBA to delete or update rows from the database audit trail (which is held in SYSTEM.AUD$ for OLS). It is possible for these users to grant the relevant privileges to other users, but it is assumed that they will not do this in practice.

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, starting 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.LCOL Whenever an audit record is written to the database audit trail, for each OLS policy that has been created for the database, a label column is present which can hold the session label.

F.AUD.LAUD The TOE will allow a suitably authorised user to enable or disable auditing of labels for a specified OLS policy.

F.AUD.LEN The TOE will allow a suitably authorised user to enable auditing of OLS events to the database audit trail for a particular OLS policy, specifying options for:

a) specific users to be audited;

b) whether auditing is BY ACCESS or BY SESSION;

c) whether events with SUCCESSFUL and/or NOT SUCCESSFUL outcomes are to be audited;

d) specific OLS events to be audited:

i. application of specified OLS policy to tables or schemas;

ii. removal of specified OLS policy from tables or schemas;

iii. the setting of user authorisations and user and program privileges;

iv. the use of all policy-specific privileges.

Note that audit records for OLS events will not be written to the audit trail unless the AUDIT_TRAIL initialisation parameter has been set to DB or OS in the database's parameter file prior to starting up the database.

F.AUD.LDIS The TOE will allow a suitably authorised user to disable auditing of OLS events to the database audit trail for a particular OLS policy, specifying options for:

a) specific users not to be audited;

b) specific OLS events not to be audited:

i. application of specified OLS policy to tables or schemas;

ii. removal of specified OLS policy from tables or schemas;

iii. the setting of user authorisations and user and program privileges;

iv. the use of all policy-specific privileges.

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

F.AUD.LVIEW The TOE allows a suitably authorised user to create a view of the audit trail which contains the specified policy's label column as well as all the entries in the audit trail written on behalf of the policy.

F.AUD.FULL With DBMS auditing, if the tablespace containing the audit trail table becomes full, no further auditable actions can occur until space is made available.

Security Mechanisms and Techniques

When authentication is performed by Oracle9*i*, a password is used for authentication. The TOE employs a one-way encryption algorithm (modified Data Encryption Standard (DES)) to encrypt passwords prior to storing them in the database. The TOE password management functions (together called the PWD mechanism) provide a Strength of Function level of *SOF-high*. This exceeds the DBMS PP Strength of Function level of *SOF-medium* [DPP].

Specific SFs supporting the claimed SOF are:

- F.IA.DBA (SOF-High); *and*
- F.IA.PWD, F.IA.ATT & F.IA.USE support F.IA.DBA by providing password

management mechanisms.

Assurance Measures

The target assurance level is EAL4 augmented with ALC_FLR.3, which exceeds the assurance requirement of EAL3 as stated in [DPP]. No other specific assurance measures are claimed. The following table identifies the Oracle9*i* documentation that supports each security assurance requirement for EAL4 and also the assurance requirement for ALC_FLR.3.

Table 8: Oracle9i Assurance Measures

Component	Name	Documents
ACM_AUT.1	Partial CM Automation	[CM]
ACM_CAP.4	Generation Support and Acceptance Procs	[CM]
ACM_SCP.2	Problem Tracking CM Coverage	[CM]
ADO_DEL.2	Detection of Modification	[OQM]
ADO_IGS.1	Installation, Generation, and Startup	[ICG] [OLS_IN] [OLS_ECD]
ADV_FSP.2	Fully Defined External Interfaces	[ERR] [OCI]
ADV_HLD.2	Security Enforcing High-level Design	[AD] [OLS_AD]
ADV_IMP.1	Subset of the TSF Implementation	[SRC] [OLS_SRC]
ADV_LLD.1	Descriptive Low-level Design	[DD] [OLS_DD]
ADV_RCR.1	Informal Correspondence Demonstration	[AD] [OLS_AD] [DD] [OLS_DD] [DT] [SRC]
ADV_SPM.1	Informal TOE Security Policy Model	[OLS_SPM]
AGD_ADM.1	Administrator Guidance	[OLS_ECD] [GA] [OLS_GA] and Oracle publications relevant to administrators
AGD_USR.1	User Guidance	[GA] [OLS_GA] and Oracle publications relevant to users
ALC_DVS.1	Identification of Security Measures	[SODE]
ALC_LCD.1	Developer Defined Life Cycle Model	[LCS]
ALC_TAT.1	Well Defined Development Tools	[CM]
ATE_COV.2	Analysis of Coverage	[TP] [OLS_TP]

Table 8: Oracle9i Assurance Measures

Component	Name	Documents
ATE_DPT.1	Testing - High-level Design	[TP] [OLS_TP]
ATE_FUN.1	Functional Testing	[TP] [OLS_TP]
AVA_MSU.2	Validation of Analysis	[GA] [OLS_GA]
AVA_SOF.1	Strength of TOE Security Functions	[SOF]
AVA_VLA.2	Independent Vulnerability Analysis	[VA] [OLS_VA]
ALC_FLR.3	Systematic Flaw Remediation	[LCS]

This Page Intentionally Blank

CHAPTER

7 Protection Profile Claims

PP Reference

The TOE conforms to the Database Management System Protection Profile (DBMS PP) [DPP].

PP Tailoring

Table 3 in chapter 5 identifies each SFR for this Security Target that was derived from [DPP] and the tailoring operations performed relative to [DPP]. The tailoring is identified in *ITALICISED CAPITAL LETTERS* within the text of each SFR in chapter 5. All of the tailoring operations are in conformance with the assignments and selections in [DPP].

PP Additions

There are additional threats, organisational security policies, and objectives included in this security target which were not in [DPP]. These are related to label-based access control and are: T.LBAC, P.LABEL, P.INFOFLOW and O.ACCESS.LBAC.

A reference to [OLS_ECD] has been added to the assumption A.TOE.CONFIG. This does not change the meaning of the assumption, rather it points to a TOE-specific document where the evaluated configuration is defined.

There is an additional underlying system assumption, A.MIDTIER, which is included to ensure accountability in multi-tier environments. Although the O-RDBMS can audit the actions of a proxy user, accountability relies upon the correct identity of the client (given during the connection by the middle-tier). As explained in chapter 1 (TOE Overview), this type of environment is an addition to the scope of evaluation (which was first introduced for Oracle8i).

An additional personnel assumption, A.USERS, has been added for label-based access control to ensure that users are assigned label authorisations and policy privileges commensurate with the degree of trust placed in them by the organisation that owns,

or is responsible for, the information processed by or stored in the TOE.

Table 3 in chapter 5 identifies each SFR for this Security Target that was not included in [DPP] (via a "*" after the component identifier).

The assurance requirements specified in this security target are those for EAL4 augmented with ALC_FLR.3. This includes all assurance requirements in [DPP] (which mandates EAL3).

CHAPTER

8 Rationale

Security Objectives Rationale

This section is required to demonstrate why the identified security objectives are suitable to counter the identified threats and meet the stated security policies.

The threats for the TOE are as per [DPP, 3.2] with the addition of T.LBAC. The OSPs for the TOE are as per [DPP, 3.3] with the addition of P.LABEL and P.INFOFLOW. The TOE security objectives are as per [DPP, 4.1] with the addition of O.ACCESS.LBAC. The environmental security objectives are as per [DPP, 4.2] with the addition of O.USERS.

[DPP 6.1] demonstrates why the security objectives identified in [DPP, 4.1] and [DPP, 4.2] are suitable to counter the threats identified in [DPP, 3.2] and meet the security policies stated in [DPP, 3.3].

The rationales for T.LBAC, P.LABEL and P.INFOFLOW are given below.

T.LBAC Rationale

T.LBAC (*Unauthorised Access to Labelled Information*) is directly countered by O.ACCESS.LBAC, which ensures that labels are provided for objects and subjects and uses these labels to enforce an information flow control policy. O.ACCESS.RESIDUAL ensures access is prevented to residual information held in memory or reused database objects. O.I&A.TOE provides support by providing the means of identifying the user attempting to access a database object. O.ACCESS.CONTROL and O.ADMIN.TOE provide support by controlling access to database control data and administrative functionality that might otherwise enable circumvention of database object access controls. O.USERS counters this threat by ensuring that administrators assign appropriate label authorisations and policy privileges to users.

P.LABEL Rationale

P.LABEL is directly satisfied by O.ACCESS.LBAC, which requires provision of labels for subjects and database objects as defined by P.LABEL. O.USERS supports this OSP by ensuring that administrators assign appropriate label authorisations and policy privileges to users in accordance with P.LABEL c).

P.INFOFLOW Rationale

P.INFOFLOW is directly satisfied by O.ACCESS.LBAC, which requires provision of

an information flow control policy as defined by P.INFOFLOW. O.USERS supports this OSP by ensuring that administrators assign appropriate label authorisations and policy privileges to users.

Assumptions Rationale

The assumptions rationale in [DPP, 6.5] applies to the TOE, with the exception that A.TOE.CONFIG has been slightly modified relative to [DPP, 3.4.1], so a modified rationale has been given below. In addition, assumptions A.MIDTIER and A.USERS are not present in [DPP] and therefore rationales have been supplied below.

A.TOE.CONFIG is directly provided by O.INSTALL part a) because [OLS_ECD] is part of the operational documentation of the TOE.

A.MIDTIER states that any middle-tier must pass the original client ID through to the TOE. A.MIDTIER is directly provided by O.INSTALL part a) because [OLS_ECD] includes this requirement for the use of a middle-tier.

A.USERS is directly satisfied by O.USERS which ensures that the users are assigned label authorisations and policy privileges commensurate with the degree of trust placed on them by the organisation that owns or is responsible for the information processed by or stored in the TOE.

Security Requirements Rationale

The TOE security objectives are as per [DPP, 4.1] with the addition of O.ACCESS.LBAC. The TOE's SFRs are as per [DPP, 5.1 and 5.2 and 5.3] with the addition of FDP_IFC.1, FDP_IFF.2, FMT_MOF.1, FMT_MSA.1.1.2, FMT_MSA.3.1.2, and FMT_MSA.3.2.2.

Suitability of Security Requirements

[DPP, 6.2.1 and 6.3 and 6.4.1] show that the SFRs defined in [DPP, 5.1 and 5.2 and 5.3] satisfy the IT security objectives defined in [DPP, 4.1].

O.ACCESS.LBAC is directly provided by FDP_IFC.1 which defines the objects of the information control policy and FDP_IFF.2 which defines the information control policy rules. FMT_MOF.1 ensures that the behaviour of the information control policy mechanism is protected from unauthorised modification. FMT_MSA.1.1.2, FMT_MSA.3.1.2 and FMT_MSA.3.2.2 provide support for the management of the information control security attributes used in controlling access to database objects.

Thus the extra IT security objective for OLS is satisfied by the extra SFRs for OLS and each extra SFR for OLS is necessary to satisfy the extra IT security objective for OLS.

This rationale thus demonstrates the suitability of the TOE security requirements.

Dependency Analysis

The tables given in [DPP, 6.2.2 and 6.4.2] apply to the TOE, except that the table below shows the dependency analysis for the components added for OLS (note that the entry for FMT_MSA.1 in the table given in [DPP, 6.2.2] is to be interpreted to apply to FMT_MSA.1.1.1). The tables given in [DPP, 6.2.2 and 6.4.2] and the table below

show that all dependencies of functional components are satisfied.

Table 9: Functional Component Dependency Analysis

Component Reference	Component	Dependencies	Dependency Reference
24	**FDP_IFC.1**	FDP_IFF.2	25
25	**FDP_IFF.2**	FDP_IFC.1 FMT_MSA.3	24 15
26	**FMT_MOF.1**	FMT_SMR.1	18
27	**FMT_MSA.1.1.2**	FDP_IFC.1 FMT_SMR.1	24 18

For the dependency analysis of the security assurance requirements: EAL4 is a self-contained assurance package and ALC_FLR.3 has no dependencies on any other component.

Demonstration of Mutual Support

The supportive dependencies discussed in [DPP, 6.2.3] apply to the TOE. The following additional supportive dependencies exist for the TOE to prevent bypassing of and tampering with the new SFRs:

FDP_RIP.1 supports FDP_IFC.1 and FDP_IFF.2 by preventing the bypassing of these SFRs through access to reused storage objects.

FIA_UID.1 and FIA_UAU.1 support FDP_IFC.1 and FDP_IFF.2 by preventing the bypassing of these SFRs by unauthorised users.

FMT_MOF.1 provides support to FDP_IFC.1 and FDP_IFF.2 by ensuring that only authorised administrative users can modify the information flow control functions.

FMT_MSA.3 provides support to FDP_IFC.1 and FDP_IFF.2 by ensuring that objects are protected by default when newly created.

FMT_MSA.1 provides support to FDP_IFC.1 and FDP_IFF.2 by controlling the modification of object security attributes.

FPT_RVM.1 supports FDP_IFC.1 and FDP_IFF.2 by ensuring that enforcement functions are always applied to prevent bypassing of these SFRs.

FPT_SEP.1 supports FDP_IFC.1 and FDP_IFF.2 by providing separate domains to prevent tampering with these SFRs.

The new SFRs for OLS do not offer any additional support to the other SFRs.

Strength of Function Validity

The strength of function specified, SOF-*high*, exceeds the strength of function required by [DPP]. The PWD mechanism is the only TOE mechanism that is probabilistic or permutational, and has a strength of SOF-*high*. This strength of function is intended to provide enough protection against straightforward or intentional attack from threat agents having a high attack potential.

Assurance Requirements Appropriate

The target assurance level is EAL4, augmented with ALC_FLR.3, which exceeds the minimum assurance requirement of EAL3 as stated in [DPP]. See [DPP, 6.7] for fur-

ther information.

ALC_FLR.3 has been included in addition to EAL4 to cause the evaluation of the TOE's flaw remediation procedures which Oracle9*i* database users need to be in place following the release of the TOE. These procedures are required to offer continuing assurance to users that Oracle9*i* provides secure storage of and access to the data which is crucial to their enterprise's success.

To meet this requirement, the flaw remediation procedures must offer:

- the ability for TOE users to report potential security flaws to Oracle,
- the resolution and correction of any flaws with assurance that the corrections introduce no new security flaws, and
- the timely distribution of corrective actions to users.

ALC_FLR.3 is the ALC_FLR component which is at an appropriate level of rigour to cover these requirements.

TOE Summary Specification Rationale

This section demonstrates that the TOE Security Functions and Assurance Measures are suitable to meet the TOE security requirements.

TOE Security Functions Satisfy Requirements

Tables 6 and 7 of chapter 6 identify the Oracle9*i* TOE Security Functions that address each of the SFRs in chapter 5.

The table below demonstrates that for each SFR the TOE security functions are suitable to meet the SFR, and the combination of TOE security functions work together so as to satisfy the SFR:

Table 10: TOE Security Function Suitability and Binding

SFR	TOE Security Functions	Rationale
FIA_AFL.1.1	F.IA.PWD	The number of allowed failed logon attempts can be configured.
FIA_AFL.1.2	F.IA.PWD	When the configured number of failed logon attempts is reached the account is locked.
FIA_ATD.1.1	F.IA.ATT F.IA.POLICY	The data dictionary stores the required security attributes
FIA_SOS.1.1	F.IA.PWD F.LIM.CNF F.IA.USE	F.IA.PWD specifies the controls available on database secrets (passwords). These controls are implemented via profiles which are required by F.LIM.CNF. F.IA.USE allows users to change their own passwords within the limits configured by an administrator.
FIA_UAU.1.1	F.IA.PRE	F.IA.PRE maps onto FIA_UAU.1.1 and FIA_UID.1.2 directly.

Table 10: TOE Security Function Suitability and Binding

SFR	TOE Security Functions	Rationale
FIA_UAU.1.2	F.IA.PRE F.IA.DBA F.IA.OSA F.IA.CSA F.IA.CSN	F.IA.CSN and F.IA.CSA state the conditions for being able to establish a database session and hence perform TSF-mediated actions. These security functions depend directly on F.IA.DBA and F.IA.OSA. F.IA.PRE is relevant because one of the actions allowed prior to session creation is attempting to establish a session.
FIA_UID.1.1	F.IA.PRE	F.IA.PRE satisfies FIA_UAU.1.1 and FIA_UID.1.1 directly.
FIA_UID.1.2	F.IA.PRE F.IA.UID F.IA.DBA F.IA.OSA F.IA.IDE F.IA.CSA F.IA.CSN	F.IA.CSN and F.IA.CSA state the conditions for being able to establish a database session and hence perform TSF-mediated actions. These security functions depend directly on F.IA.DBA and F.IA.OSA. F.IA.PRE is relevant one of the actions allowed prior to session creation is attempting to establish a session. F.IA.IDE ensures that the identity of the user is known for the duration of the session, once created.
FIA_USB.1.1	F.IA.ATT F.IA.POLICY F.IA.SESSION F.IA.SESSUPD F.APR.EDR F.PRI.SPRIV F.PRI.PPRIV F.LBAC.LABUPD F.PRI.XVP F.LBAC.XVP F.LBAC.TRIG-GER F.PRI.PRX	F.IA.ATT and F.IA.POLICY provide the security attributes for each user. The effective security attributes for a database session are controlled by F.IA.SESSION, F.IA.SESSUPD, F.APR.EDR, F.PRI.SPRIV, F.PRI.PPRIV and F.LBAC.LABUPD. In addition, F.PRI.XVP and F.LBAC.XVP define security attributes associated with views and program units which act on behalf of the owning user. F.LBAC.TRIGGER defines security attributes associated with triggers which act on behalf of the triggering user. F.PRI.PRX defines security attributes associated with proxy user sessions.
FDP_ACC.1.1	F.ACCESS F.DAC.OBID F.DAC.OBREF F.DAC.SUA F.DAC.OBA	F.DAC.OBID and F.DAC.OBREF ensures that all objects (which are subject to DAC) can be uniquely identified. F.ACCESS, F.DAC.SUA and F.DAC.OBA state that the DAC policy extends to all subjects and objects.
FDP_ACF.1.1	F.DAC.OBID F.DAC.OBREF F.DAC.SUA F.DAC.OBA F.DAC.POL F.PRI.SPRIV	F.DAC.OBID and F.DAC.OBREF ensure that all objects (which are subject to DAC) can be uniquely identified. F.DAC.SUA includes the subject and their enabled privileges (as specified in F.PRI.SPRIV) in the DAC policy. F.DAC.OBA states that the object and any associated object privileges are considered by the DAC policy. F.DAC.POL is a statement of the DAC policy.
FDP_ACF.1.2	F.IA.CNF F.DAC.OBID F.DAC.OBREF F.DAC.POL F.PRI.SPRIV F.AUD.CNF	F.DAC.POL a) and b) specifies access to objects based on ownership or object privileges. F.DAC.OBID and F.DAC.OBREF are relevant as they define object ownership which is the basis of the DAC policy. FPRI.SPRIV is relevant as it defines which privileges are enabled for any user. F.IA.CNF and F.AUD.CNF are relevant I&A data and the audit trail are subject to the DAC policy.
FDP_ACF.1.3	F.DAC.POL F.PRI.SPRIV F.AUD.CNF	F.DAC.POL c) specifies access to objects based on enabled system privileges. F.DAC.POL d) and e) cover access via connections AS SYSDBA and AS SYSOPER. F.SPRIV is relevant as it defines which privileges are enabled for any user. F.AUD.CNF is relevant as the audit trail is subject to the DAC policy

Table 10: TOE Security Function Suitability and Binding

SFR	TOE Security Functions	Rationale
FDP_ACF.1.4	N/A	This SFR does not mandate any functionality. It is included for compliance with the CC.
FDP_RIP.2.1	F.DAC.OR	F.DAC.OR satisfies FDP_RIP.2.1 directly.
FMT_MSA.1.1.1	F.APR.GOP F.APR.ROP F.APR.GRSP F.APR.GRR	F.APR.GOP and F.APR.ROP cover FMT_MSA.1.1.1 a) which is concerned with modifying object privileges. F.APR.GRSP covers FMT_MSA.1.1.1 b) which is concerned with modifying system privileges. F.APR.GRR covers FMT_MSA.1.1.1 c) which is concerned with modifying roles.
FMT_MSA.3.1.1	F.DAC.POL F.PRI.SPRIV	F.DAC.POL and F.PRI.SPRIV implicitly include restrictive default values. If a user has not been explicitly granted the necessary privilege or a role containing the required privilege then the requested action will not succeed.
FMT_MSA.3.2.1	F.DAC.OBA F.DAC.POL F.APR.GOP F.APR.GRSP F.APR.GRR	Unless access to an object has been explicitly granted, as described in F.DAC.OBA, F.APR.GOP, F.APR.GRSP and F.APR.GRR, no access will be allowed. On object creation no object privileges are granted and it is not possible to configure this to be the case. F.DAC.POL is relevant as it enforces the database object access SFP.
FMT_MTD.1.1	F.IA.ATT F.IA.POLICY F.LIM.CNF F.APR.GOP F.APR.ROP F.APR.GRSP F.APR.GRPP F.APR.GRR F.AUD.ACC F.AUD.DEL F.LBAC.LABSET F.LBAC.LABUPD	These TOE security functions are concerned with the modification of TSF data (security attributes and audit data). This data is stored in the data dictionary and is protected from unauthorised access by the same mechanism as all other data in the database. F.IA.ATT, F.IA.POLICY and F.LIM.CNF cover identification and authentication data and user label authorisations and resource limit attributes. F.APR.* cover privilege and role TSF data. F.AUD.* cover audit data. F.LBAC.LABSET and F.LBAC.LABUPD cover the setting and updating of object labels.
FMT_REV.1.1	F.LIM.CNF F.APR.ROP F.APR.GRSP F.APR.GRPP F.APR.GRR	Only suitably privileged users can revoke (or modify) the following attributes: resource limits (F.LIM.CNF), object privileges (F.APR.ROP), system privileges (F.APR.GRSP), policy privileges (F.APR.GRPP) and roles (F.APR.GRR).
FMT_REV.1.2	F.PRI.SPRIV	Directly granted privileges and roles are revoked immediately. This is more rigorous than SFR FMT_REV.1.2. Revocation of roles takes effect when a role is re-enabled in the current session or a new user session is created.
FMT_SMR.1.1	F.IA.UID F.IA.CSA F.IA.CSN F.APR.GRR	F.IA.UID, F.IA.CSA and F.IA.CSN in combination ensure that the TSF maintains normal database users and database administrative users. F.APR.GRR covers database roles defined by a suitably authorised user.
FMT_SMR.1.2	F.IA.CSA F.APR.DER F.APR.EDR F.PRI.PRX	F.APR.DER and F.APR.EDR cover granting database roles to database users. F.IA.CSA is relevant because it specifies how to allow a user to connect AS SYSDBA or AS SYSOPER. F.PRI.PRX covers database roles available to a proxy user session.

Table 10: TOE Security Function Suitability and Binding

SFR	TOE Security Functions	Rationale
FPT_RVM.1.1	F.IA.IDE F.ACCESS F.DAC.POL F.LBAC.POL	F.IA.IDE ensures that the TOE always knows who the current user is. F.ACCESS, F.DAC.POL and F.LBAC.POL ensure that the database access control policy enforcement functions are always invoked for this user.
FPT_SEP.1.1	F.IA.IDE F.DAC.SEP	F.IA.IDE ensures that the identity of the user associated with each interaction with the TOE is clear. F.DAC.SEP ensures that the interactions between different users and the TOE cannot interfere with each other. Additionally there is no way to access the TOE except through the evaluated interfaces described by the TOE security functions.
FPT_SEP.1.2	F.IA.IDE F.DAC.SEP	F.IA.IDE ensures that the identity of the user associated with each interaction with the TOE is clear. F.DAC.SEP ensures that the interactions between different users and the TOE cannot interfere with each other.
FRU_RSA.1.1	F.LIM.CNF F.LIM.POL F.LIM.NSESS F.LIM.TIME F.LIM.RSESS F.LIM.RCALL	F.LIM.CNF covers configuration of the resource quotas. F.LIM.POL, F.LIM.NSESS, F.LIM.TIME, F.LIM.RSESS and F.LIM.RCALL enforces the resource quotas configured.
FTA_MCS.1.1	F.LIM.NSESS	F.LIM.NSESS directly satisfies FTA_MCS.1.2
FTA_MCS.1.2	F.LIM.NSESS F.LIM.POL	As with FTA_MCS.1.1 except that F.LIM.POL ensures that the default number of concurrent sessions allowed is enforced if a user specific configuration has not been specified.
FTA_TSE.1.1	F.IA.CSN F.IA.CSA	F.IA.CSN and F.IA.CSA define the pre-requisites for session establishment, including possession of the CREATE SESSION privilege and being identified as SYSDBA/SYSOPER, respectively. These are configured on the basis of individual user identity. Therefore, it is possible to deny access based on user identity.
FAU_GEN.1.1	F.AUD.SOM F.AUD.SEV F.AUD.ALW F.AUD.LAUD F.AUD.LEN F.AUD.LDIS	The database audit functionality is always active. Whether or not auditing is actually performed is dependent on the configuration of a parameter in the init.ora file which is controlled by the OS. F.AUD.SOM, F.AUD.SEV, F.AUD.ALW, F.AUD.LAUD, F.AUD.LEN and F.AUD.LDIS ensure all actions configured to be audited are audited.
FAU_GEN.1.2	F.AUD.INF F.AUD.LCOL	F.AUD.INF and F.AUD.LCOL satisfy FAU_GEN.1.2
FAU_GEN.2.1	F.AUD.INF	F.AUD.INF directly satisfies FAU_GEN.2.1
FAU_SAR.1.1	F.AUD.ACC	F.AUD.ACC directly satisfies FAU_SAR.1.1
FAU_SAR.1.2	F.AUD.VIEW F.AUD.LVIEW	F.AUD.VIEW and F.AUD.LVIEW satisfy FAU_SAR.1.2

Table 10: TOE Security Function Suitability and Binding

SFR	TOE Security Functions	Rationale
FAU_SAR.3.1	F.AUD.VIEW F.AUD.LVIEW F.AUD.ACC	F.AUD.VIEW and F.AUD.LVIEW satisfy FAU_SAR.3.1. Additionally F.AUD.ACC determines which records are available to the user for selection.
FAU_SEL.1.1	F.AUD.SOM F.AUD.SEV F.AUD.ALW F.AUD.CNF	F.AUD.SEV and F.AUD.CNF allow a suitably privileged user to configure exactly which events should be audited. F.AUD.SOM and F.AUD.ALW specify events that are always audited. Note that for audit records database subjects are always the database users, so that for example an audit record generated by a stored procedure will be generated with the username of the invoker, not that of the procedure or the procedure owner.
FAU_STG.1.1	F.AUD.DEL	F.AUD.DEL directly satisfies FAU_STG.1.1.
FAU_STG.1.2	F.AUD.DEL	F.AUD.DEL protects audit records from unauthorised modification or deletion.
FAU_STG.4.1	F.AUD.FULL	F.AUD.FULL directly satisfies FAU_STG.4.1
FDP_IFC.1.1	F.ACCESS F.LBAC.POL F.LBAC.REF	F.ACCESS ensures that the LBAC access control policy defined via F.LBAC.POL and F.LBAC.REF is enforced.
FDP_IFF.2.1	F.LBAC.POL	F.LBAC.POL defines the LBAC policy, which is based on database subject and database object labels.
FDP_IFF.2.2	F.LBAC.POL	F.LBAC.POL defines the LBAC policy.
FDP_IFF.2.3	F.IA.SESSUPD	F.IA.SESSUPD satisfies FDP_IFF.2.3 directly.
FDP_IFF.2.4	F.LBAC.XVP	F.LBAC.XVP satisfies FDP_IFF.2.4 directly.
FDP_IFF.2.5	F.LBAC.POL F.PRI.PPRIV	F.LBAC.POL b) ensures that appropriate policy privileges can be used to override LBAC access mediation rules. F.PRI.PPRIV defines when policy privileges come into effect.
FDP_IFF.2.6	N/A	This SFR does not mandate any functionality. It is included for compliance with the CC.
FDP_IFF.2.7	F.LBAC.POL	F.LBAC.POL is partly based on the dominance ordering relationship.
FMT_MOF.1.1	F.LBAC.MOD	F.LBAC.MOD satisfies FMT_MOF.1.1 directly.
FMT_MSA.1.1.2	F.IA.POLICY F.APR.GRPP F.LBAC.LABUPD	F.IA.POLICY covers the updating of user security attributes and F.APR.GRPP specifically covers updating policy privileges. F.LBAC.LABUPD ensures that LBAC rules are applied to the updating of object labels.
FMT_MSA.3.1.2	F.LBAC.LABSET	F.LBAC.LABSET ensures that the LBAC rules are applied so that an initial value for the label is always provided when a database object is created (i.e. no system default is used).

Table 10: TOE Security Function Suitability and Binding

SFR	TOE Security Functions	Rationale
FMT_MSA.3.2.2	F.LBAC.LABSET	F.LBAC.LABSET ensures that the LBAC rules are always applied to the provision of the initial value of the label when a database object is created (hence no user can cause an initial label value to be assigned which violates the LBAC rules).

PP Claims Rationale

Chapter 5 lists all of the SFRs included in this security target; this list includes all of the SFRs identified in the DBMS PP. All of the operations applied to the SFRs derived from the DBMS PP are in accordance with the requirements of the DBMS PP.

Table 8 in Chapter 6 demonstrates that all assurance requirements are suitably met by one or more assurance measures.

This Page Intentionally Blank

ANNEX

A References

[AD] *Oracle9i Architecture Release 2 (9.2.0),* Oracle Corporation.

[ADG] *Oracle9i Application Developer's Guide - Fundamentals, Release 2 (9.2),* Oracle Corporation.

[CC] *Common Criteria for Information Technology Security Evaluation,* Version 2.1, ISO/IEC 15408, CCIB-99-031, 032 & 033, August 1999.

[CEM_FLR] *Common Methodology for Information Technology Security Evaluation,* Part 2: Evaluation Methodology, Supplement: ALC_FLR - Flaw Remediation Version 1.1, CEM-2001/0015R, February 2002.

[CM] *Oracle9i Configuration Management Plan Release 2 (9.2.0),* Oracle Corporation.

[CON] *Oracle9i Database Concepts, Release 2 (9.2),* Oracle Corporation.

[DAG] *Oracle9i Database Administrator's Guide, Release 2 (9.2),* Oracle Corporation.

[DD] *Oracle9i Detailed Design Release 2 (9.2.0),* Oracle Corporation.

[DPP] *Database Management System Protection Profile (DBMS PP),* Issue 2.1, Oracle Corporation, May 2000.

[DT] *Oracle9i Design Traceability, Release 2 (9.2.0),* Oracle Corporation.

[ERR] *Oracle9i Database Error Messages, Release 2 (9.2),* Oracle Corporation.

[FIPS46-3] *Federal Information Processing Standard Publication 46-3,* National Institute of Standards and Technology (NIST), October 1999.

[FIPS81] *Federal Information Processing Standard Publication 81,* National Institute of Standards and Technology (NIST), December 1980.

[GA] *Oracle9i Administrator and User Guidance Analysis Release 2 (9.2.0),* Oracle Corporation.

[ICG] *Oracle9i Installation and Configuration Guide, Release 2 (9.2),* Oracle Corporation.

[ITSEC] *Information Technology Security Evaluation Criteria,* Issue 1.2, Commission of the European Communities, 28 June 1991.

[LBACFS] *Functional Specification for Label-based Access Controls,* Oracle Corporation.

[LCS] *Life Cycle Support for Oracle9i, Release 2 (9.2.0),* Oracle Corporation.

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

[OCI] *Oracle Call Interface Programmers Guide, Release 2 (9.2),* Oracle Corporation.

[OLS_AD] *OLS Architecture for Oracle9i Release 2 (9.2.0),* Oracle Corporation.

[OLSAG] *Oracle Label Security Administrator's Guide, Release 2 (9.2),* Oracle Corporation.

[OLS_DD] *OLS Detailed Design for Oracle9i Release 2 (9.2.0),* Oracle Corporation.

[OLS_ECD] *OLS Evaluated Configuration Document for Oracle9i Release 2 (9.2.0),* Oracle Corporation.

[OLS_GA] *OLS Administrator and User Guidance Analysis for Oracle9i Release 2 (9.2.0),* Oracle Corporation.

[OLS_IN] *Oracle Label Security Installation Notes, Release 9.2,* Oracle Corporation.

[OLS_SPM] *OLS Security Policy Model for Oracle9i Release 2 (9.2.0)*, Oracle Corporation.

[OLS_SRC] *OLS Source Code for Oracle9i Release 2 (9.2.0),* Oracle Corporation.

[OLS_ST8i] *OLS Security Target for Oracle8i Release 3 (8.1.7)*, Oracle Corporation.

[OLS_TP] *OLS Test Plan, Procedures, Results, and Analysis for Oracle9iRelease 2 (9.2.0),* Oracle Corporation.

[OLS_VA] *OLS Vulnerability Analysis for Oracle9i, Release 2 (9.2.0),* Oracle Corporation.

[OQM] *Quality Manual for Manufacturing & Distribution,* Oracle Corporation.

[PLS] *PL/SQL User's Guide and Reference, Release 2 (9.2),* Oracle Corporation.

[SAPAFS] *Functional Specification for Secure Access Policy Adapter,* Oracle Corporation.

[SODE] *Security of the Oracle Development Environment,* Oracle Corporation.

[SOF] *Strength of Functions Analysis,* Oracle Corporation.

[SQL] *Oracle9i SQL Reference*, *Release 2 (9.2),* Oracle Corporation.

[SQL92] *Database Language SQL, ISO/IEC 9075:1992 and ANSI X3.135-1992,*

[SRC] *Oracle9i Source Code, Release 2 (9.2.0),* Oracle Corporation.

[SRF] *Oracle9i Server Reference*, *Release 2 (9.2),* Oracle Corporation.

[ST8i] *Oracle8i Release 3 (8.1.7) Security Target*, Oracle Corporation.

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

[TP] *Oracle9i Test Plan, Procedures, Results, and Analysis,Release 2 (9.2.0),* Oracle Corporation.

[VA] *Oracle9i Vulnerability Analysis,Release 2 (9.2.0),* Oracle Corporation.

This Page Intentionally Blank

ANNEX

B Glossary

Acronyms

DAC	Discretionary Access Control
DDL	Data Definition Language
DES	Data Encryption Standard
DML	Data Manipulation Language
LBAC	Label-Based Access Control
OLS	Oracle Label Security
O-RDBMS	Object-Relational Database Management System
SF	Security Function
SFP	Security Function Policy
SFR	Security Functional Requirement
SOF	Strength of Function
SQL	Structured Query Language
TOE	Target Of Evaluation
TSC	TOE Scope of Control

TSF	TOE Security Functions
TSFI	TSF Interface
TSP	TOE Security Policy

Terms

Authorised administrative user	Another name for a Database Administrative User.
Data Definition Language (DDL)	The SQL statements used to define the schema and schema objects in a database [SQL]
Data dictionary	A set of internal Oracle tables that contain information about the logical and physical structure of the database. [SCN]
Data Encryption Standard (DES)	A standard for encryption, FIPS PUB 46-3 and FIPS PUB 81. [FIPS46-3],[FIPS81]
Data Manipulation Language (DML)	The SQL statements used to query and manipulate data in schema objects [SQL]
Data server	A component of a DBMS that supports concurrent access to a database by multiple users, possibly at different nodes in a distributed environment. [ST]
Database	A collection of data that is treated as a unit; the general purpose of a database is to store and retrieve related information [SCN]
Database administrative user	A database user to whom one or more administrative privileges have been granted. [DPP] This includes users connected AS SYSOPER or AS SYSDBA as well as Normal Users who are authorised to perform an administrative task via the posession of an administrative privilege which permits the operation of the task.
Database connection	A communication pathway between a user and a DBMS. [DPP]
Database link	A definition of a one-way communication path from an Oracle database to another database. [SCN]
Database non-administrative user	A database user who only has privileges to perform operations in accordance with the TSP. [DPP]
Database object	An object contained within a database. [DPP]
Database session	A connection of an identified and authenticated user to a specific database; the session lasts from the time the user connects (and is identified and authenticated) until the time the user disconnects. [DPP]

Database subject	A subject that causes database operations to be performed. [DPP]
Database user	A user who interacts with a DBMS and performs operations on objects stored within the database. [DPP]
Discretionary Access Control	Access control based on access rights granted by users other than the System Security Officer. [MEMO 1]
Instance	The combination of a set of Oracle background processes and memory that is shared among the processes. A database instance must be started (the shared memory allocated and the background processes created) by an authorised administrative user before the database managed by the instance can be accessed. [SCN]
Interface product	A TOE component that resides in a user process and can be used to communicate with an Oracle database server in a secure manner. [ST]
Label-Based Access Control	This type of access control is based on access rights granted by the system administrator. The administrator chooses which data in the database are to be protected by Label-Based Access Control according to OLS policies which he or she defines. The administrator uses these OLS policies to control the allocation of labels to objects to reflect their sensitivity. The administrator provides users with authorisations to permit access to an appropriate subset of the labelled data. [OLSAG]
LBAC administrator	A user who is able to create, alter and drop OLS policies in the database by virtue of possessing the LBAC_DBA role and `EXECUTE` privilege on the SA_SYSDBA package.
Normal User	A database user who has made a normal connection to the database. This can include the users SYS and SYSTEM but excludes users connected AS SYSOPER or AS SYSDBA.
Object	An entity within the TSC that contains or receives information and upon which subjects perform operations. Objects are visible through the TSFI and are composed of one or more TOE resources encapsulated with security attributes. [CC]
Object-Relational Database Management System (ORDBMS)	A DBMS that supports object-oriented technology as well as relational databases. [SCN]
OLS Policy	OLS policies are established by LBAC administrators and OLS policy administrators to specify how Label-Based Access Control is to be enforced on a database. [OLSAG]
OLS Policy administrator	A user who is able to execute the administrative packages for the OLS policy for which they also possess the corresponding *policy*_DBA role.
Owner	The owner of a named database object is the database user who is responsible for the object and may grant other database users access to the object on a discretionary basis. [DPP]
Platform	The combination of software and hardware underlying the DBMS. [ST]
Privilege	A right to access objects and/or perform operations that can be granted to some users and not to others. [DPP]

Privilege, database administrative	A privilege authorising a subject to perform operations that may bypass, alter, or indirectly affect the enforcement of the TSP. [DPP]
Privilege, database object access	A privilege authorising a subject to access a named database object. [DPP]
Privilege, directly granted	An Oracle system or object privilege that has been explicitly granted to a user. Privileges granted to any roles the user has been granted are not included in the set of directly granted privileges. [SCN]
Privilege, object	An Oracle privilege that allows users to perform a particular action on a specific schema object. Oracle object privileges are database object access privileges. [SCN]
Privilege, policy	Administrators give policy privileges to a user or stored program unit to allow aspects of the label-based access control policy to be bypassed. In addition, the administrator can give policy privileges to authorise the user to perform specific actions, such as the ability of one user to assume the authorisations of a different user. [OLSAG]
Privilege, system	An Oracle privilege that allows users to perform a particular system-wide action or a particular action on a particular type of object. Some Oracle system privileges are database administrative privileges. [SCN]
Program unit	A PL/SQL program; a procedure, function, or package. [PLS]
Role (CC)	A predefined set of rules establishing the allowed interactions between a user and the TOE. [CC]
Role (Oracle)	A named group of related system and/or object privileges that can be granted to users or to other roles. [SCN]
Schema	A collection of logical structures of data (schema objects), owned by a specific database user. [SQL]
Security attribute	Information associated with subjects, users, and/or objects which is used for the enforcement of the TSP. [CC]
Security domain	The set of objects that a subject has the ability to access. [TCSEC]
Security Function (SF)	A part or parts of the TOE which have to be relied upon for enforcing a closely related subset of the rules from the TSP. [CC]
Security Function Policy (SFP)	The security policy enforced by a SF. [CC]
Security Functional Requirement (SFR)	A security functional requirement defined in a protection profile or security target. [CC]
Server process	An Oracle process that services requests for access to an Oracle database from connected user processes. [SCN]
Session label	When the administrator sets up the user label authorisations for the user, he or she also specifies the user's initial session label. The session label is the particular combination

	of level, compartments, and groups at which a user works at any given time. The user can change the session label provided that it remains within the user's label authorisations. [OLSAG]
SOF-high	A level of the TOE strength of function where analysis shows that the function provides adequate protection against deliberately planned or organised breach of TOE security by attackers possessing a high attack potential. [CC]
SQL statement	A string of SQL text containing a command and supporting clauses. All access to an Oracle database is via SQL statements. [SCN]
Strength of Function (SOF)	A qualification of a TOE security function expressing the minimum efforts assumed necessary to defeat its expected security behaviour by directly attacking its underlying security mechanisms. [CC]
Structured Query Language (SQL)	A standardised database access language; Oracle8 SQL is a superset of the ANSI/ISO SQL92 standard at entry level conformance. [SQL]
Subject	An entity within the TSC that causes operations to be performed. [CC]
Suitably authorised user	A user who is authorised to perform an administrative task via the posession of an administrative privilege which permits the operation of the task. This includes users connected AS SYSOPER or AS SYSDBA as well as privileged Normal Users.
System	A specific IT installation, with a particular purpose and operational environment [CC]
Target Of Evaluation (TOE)	The product or system being evaluated. [CC]
TOE resource	Anything usable or consumable in the TOE. [CC]
TOE Scope of Control (TSC)	The set of interactions which can occur with or within a TOE and are subject to the rules of the TSP. [CC]
TOE Security Functions (TSF)	A set consisting of all the software of the TOE that must be relied on for the correct enforcement of the TSP. [CC]
TOE Security Policy (TSP)	A set of rules that regulate how assets are managed, protected and distributed within a TOE. [CC]
TSF Interface (TSFI)	A set of interfaces, whether interactive (man-machine interface) or programmatic (application programming interface), through which TOE resources are accessed, mediated by the TSF, or information is obtained from the TSF. [CC]
User	Any entity (human or machine) outside the TOE that interacts with the TOE. [CC]
User Label Authorisations	Each user authorised to access data protected by a given OLS policy has *user label authorisations* which include a maximum and minimum level, a set of authorised compartments, a set of authorised groups, and, for each compartment and group, a specification of read-only access, or read-write access. [OLSAG]
User process	A process that requests services, on behalf of a user or application, from an Oracle server process. [SCN]

This Page Intentionally Blank