Teradata® Relational Database Management System

Version 2, Release 5.0.2

Security Target (Version 1.0)

11 October 2004

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1 SECURITY TARGET IDENTIFICATION

This section provides identifying information for the Teradata® Relational Database Management System (RDBMS) Version 2, Release 5.0.2 Security Target (ST), by identifying information regarding the Target of Evaluation (TOE).

1.2 SECURITY TARGET NAME

Teradata® Relational Database Management System Version 2, Release 5.0.2 Security Target, Version 1.0, 11 October 2004

Security Target Author: Booz Allen Hamilton

1.3 TOE IDENTIFICATION

The TOE defined in this ST is Teradata® RDBMS Version 2, Release 5.0.2 (V2R5.0.2). The TOE is a Relational Database Management System and may be interchangeably referred to as Teradata® RDBMS V2R5.0.2 or Teradata® Server within this ST.

1.4 EVALUATION STATUS

The Booz Allen Hamilton Common Criteria Testing Laboratory has evaluated this ST.

1.5 EVALUATION ASSURANCE LEVEL

Assurance claims conform to EAL2 (Evaluation Assurance Level 2) from Part 3 of the *Common Criteria for Information Technology Security Evaluation, Version 2.1* (ISO/IEC 15408:1999) incorporated with Common Criteria Interpretations Management Board (CCIMB) interpretations as of October 22, 2003.

1.6 KEYWORDS

Relational Database Management System, Parsing Engine (PE), Access Module Processor (AMP), Parallel Data Extension (PDE), Structured Query Language (SQL), Discretionary Access Control (DAC).

1.7 SECURITY TARGET OVERVIEW

This ST provides the specification for the Teradata® RDBMS V2R5.0.2 Target of Evaluation. The TOE described herein is a Relational Database Management System that provides discretionary access control to protect stored database objects and resources.

This ST describes the assumptions, threats, objectives, requirements, organizational security policies and rationale for the Teradata® RDBMS V2R5.0.2 and its IT environment. The language used in this Security Target is consistent with the *Common Criteria for Information Technology Security Evaluation, Version 2.1* (ISO/IEC 15408:1999) incorporated with CCIMB

interpretations as of October 22, 2003. In addition, the spelling of several terms used within this Security Target conforms to the internationally accepted English, not always consistent with the current U.S. English spelling norms.

1.8 SECURITY TARGET ORGANIZATION

Chapter 1 of this ST provides introductory and identifying information for the Teradata® RDBMS V2R5.0.2. Chapter 2 describes the TOE and provides some guidance on its use. Chapter 3 provides a security environment description in terms of assumptions, threats and organizational security policies. Chapter 4 identifies the security objectives of the TOE and of the Information Technology (IT) environment. Chapter 5 provides the TOE security functional requirements, the assurance requirements, as well as requirements on the IT environment. Chapter 6 is the TOE Summary Specification, a description of the functions provided by Teradata® RDBMS V2R5.0.2 to satisfy the security functional and assurance requirements. Chapter 7 provides a rationale for claims of conformance to a registered Protection Profile (PP). Chapter 8 provides a rationale, or pointers to rationale, for objectives, requirements, TOE Summary Specification, and PP claims.

1.9 COMMON CRITERIA CONFORMANCE

The TOE is compliant with the Common Criteria (CC) Version 2.1, functional requirements (Part 2) extended and assurance requirements (Part 3) conformant for EAL2 incorporated with CCIMB interpretations as of October 22, 2003.

1.10 PROTECTION PROFILE CONFORMANCE

The TOE does not claim Protection Profile conformance.

2. TOE DESCRIPTION

The TOE is the Teradata® RDBMS V2R5.0.2. The TOE accesses, stores, and operates on data using Teradata® Structured Query Language (Teradata® SQL), which is compatible to ANSI SQL with extensions. The TOE was developed to allow users to view and manage large amounts of data as a collection of related tables.

The Teradata® RDBMS V2R5.0.2 enforces a discretionary access control policy such that the owner of a database object or resource (databases, tables, views, stored procedures and macros) has the authority to permit an authorized user access to those objects or resources.

2.1 TERADATA â DATABASE ROLES

The Teradata® RDBMS V2R5.0.2 maintains roles for a security administrator and authorized user.

The security administrator is responsible for maintaining the user community and preventing unauthorized users from gaining access to physical resources, such as disk space and computer cycles. The security administrator can audit events on the TOE to detect any possible security hazards. If unauthorized or undesirable activity occurs, the security administrator can take action such as changing compromised passwords or access rights to address the problem. It is the security administrator's responsibility to define and implement controls effectively to maintain security.

The authorized user is capable of carrying on dialog with the system beyond the logon process in order to access data. The user submits SQL queries to obtain information from the database, and if the Teradata® RDBMS V2R5.0.2 identifies the user name as an authorized user, and if the password is correct for that user, then the database establishes a session.

The security administrator and authorized user roles will be addressed throughout the ST in further detail.

2.2 TOE DEPLOYMENT

The TOE functions as a server in a client-server IT architecture. This architecture is further described in Section 2.3 TOE Boundary.

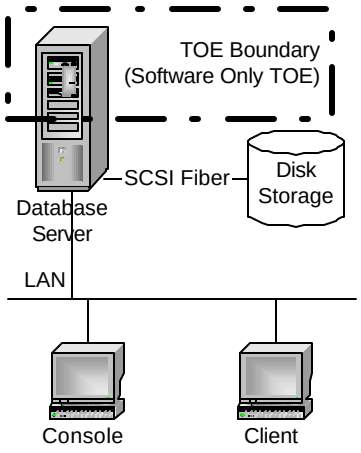
2.3 TOE BOUNDARY

This section provides information for the purpose of evaluating the TOE. This includes descriptions of the TOE physical and logical boundaries for the purpose of evaluation.

2.3.1 Physical Boundary

The physical boundary of the TOE is depicted in the following figure.

Figure 2-1 TOE Physical Boundary

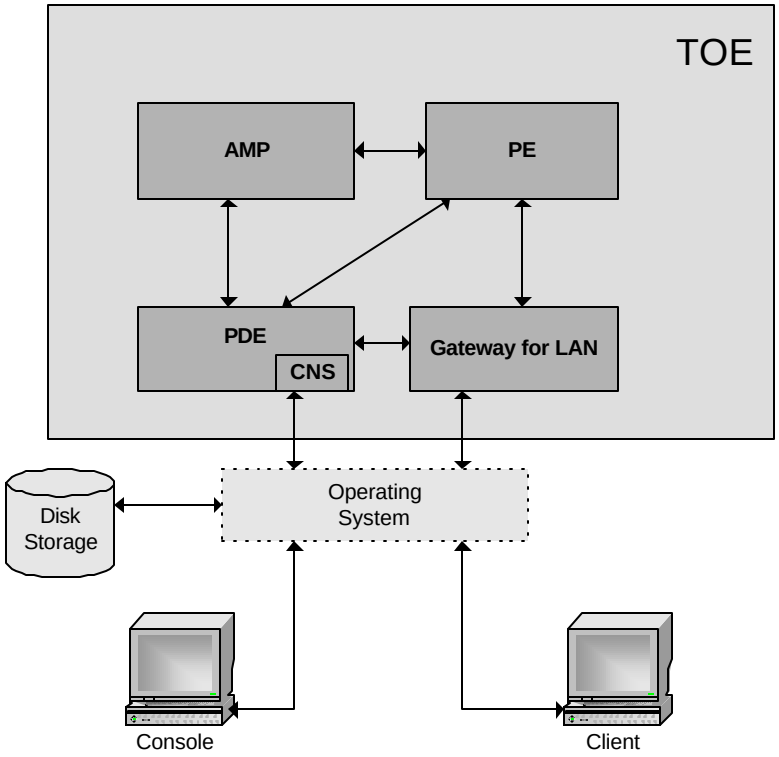


In Figure 2-1, the TOE software resides on a Database Server. The security administrator interacts with the TOE from both Console and Client nodes. The security administrator uses Teradata® SQL, which is compatible to ANSI SQL with extensions, from the Client to manage the authorized users, audit trail and functions provided by the TOE. The storage of database structures and values are retained in a Disk Storage device that is attached to the Database Server via a SCSI Fiber connection.

2.3.2 Logical Boundary

The logical boundary of the TOE includes the identification and description of the TOE software components that run on the Database Server. The TOE is comprised of several components including the Parallel Database Extension (PDE), Gateway for LAN, Parsing Engine (PE) and the Access Module Processor (AMP). The following figure identifies the logical boundary of the TOE.

Figure 2-2 TOE Logical Boundary



The software structures of the TOE are broken down and discussed in the following sub-sections.

2.3.2.1 Parallel Database Extension

In Figure 2-2, the Parallel Database Extension (PDE) element is a software interface layer that operates on top of the host Operating System (OS), thus providing an interface between the TOE and the underlying OS software. The PDE includes a BYNet Driver that manages the communication devices that interconnect the hardware nodes on which the server software is resident. It provides a standard interface for inter-process communications across nodes in a multi-node environment.

In addition, the PDE is responsible for starting all tasks defined in the PE and AMP components. Therefore, if a foreign application were executing on a server node and attempted to call a PDE function that call would be rejected. The PDE includes the following subcomponents:

- Global Objects Manager, which maintains the OS resident files that are common to all hardware nodes on which the server is resident.
- Error Logger, which inserts messages into the OS event logs.
- Monitor, which is used for synchronization between tasks.
- Segment/Memory/Disk Subsystem, which controls access to memory and disk resources by all server components.

- System Time, which synchronizes the system clock between the hardware nodes of the server and provides an interface to time functions for the server components.

- Task Allocation, which interacts with the underlying OS to create and terminate tasks to which system resources can be allocated. The task allocation manages the initial memory allocated to the tasks.

- Task Scheduler, which interacts with the underlying OS scheduler to prioritize the execution of the tasks of the server components.

The PDE also includes a Console Subsystem (CNS). The CNS manages the interface for input and output generated from a Database Window (DBW) on the Console. It provides access to the following utilities:

- Priority Scheduler for defining scheduling categories and time slices for database tasks.

- Control (ctl) for managing system options controlling operational characteristics.

- CNSRun for execution of script based database utilities.

2.3.2.2 Gateway for LAN

The function of the Gateway for LAN element is to provide the client communications interface over the LAN. It receives all messages sent from the client to the server. This includes not only messages containing Teradata® SQL statements but also messages for functions such as connecting and disconnecting sessions, determining the configuration of the server, establishing the security protocols to be used between the client and server, and responding to test messages that determine the health of the server over the LAN.

When the Gateway for LAN receives messages from the client that contain Teradata® SQL requests, it checks those messages to ensure they conform to the specified protocol. Then, those messages are forwarded to the other components within the PE. In addition, the Gateway for LAN receives response messages from the PE, and returns them to the appropriate client.

The Gateway for LAN also interacts with PDE in order to utilize OS services. It also utilizes the PDE for memory management and message handling functions.

2.3.2.3 Parsing Engine

The Parsing Engine (PE) element provides the interface between the client application and the Gateway for LAN. It is comprised of three components that include the Session Controller, Parser and Dispatcher.

2.3.2.3.1 Session Controller

The Session Controller processes external requests to establish or terminate a logical connection between the application and the server. It also provides for the recovery of sessions following client or server failures. The Session Controller manages session activities, such as logon, password validation and logoff.

2.3.2.3.2 Parser

The Parser decomposes SQL into relational data management processing steps. It processes external requests containing Teradata® SQL. Then the Parser syntactically processes the Teradata® SQL statements and prepares an execution plan to process the statement.

2.3.2.3.3 Dispatcher

The Dispatcher receives the processing steps from the Parser and sends them to the AMP. In addition, the Dispatcher monitors the completion of the steps and handles errors encountered during processing.

2.3.2.4 Access Module Processor

The Access Module Processor element consists of two components. These components include the AMP Worker Task and File System. The AMP Worker Task receives and processes the steps of an execution plan from the Dispatcher component of the Parsing Engine. The File System component maintains the disk resident structure of the relational tables managed by the TOE. The File System receives rows that are generated by the AMP Worker Task and places them in disk blocks. The File System is responsible for the creation, updating, reading and deletion of rows in relational tables.

2.4 COMPONENTS EXTERNAL TO THE TOE

The components identified in this section exist in the TOE environment and are important for discussion in how they interoperate with the TOE. These components include the Database Server Node and workstations referred to as either the Console or Client Nodes.

2.4.1 Database Server Node

The Database Server Node (also referred to as the Processor Node) serves as the hardware platform upon which the TOE software, and underlying OS operate. The Processor Node is a hardware assembly containing several, tightly coupled Central Processing Units (CPUs) and associated memory storage.

2.4.2 Disk Subsystem

The Disk Subsystem consists of the hard drive storage media and its associated SCSI interface. The Disk Subsystem receives the disk blocks that are created by the AMP Worker Task in order to achieve physical file storage.

2.4.3 Console Node

The security administrator uses the Console's Database Window (DBW) software to access the utilities and tools for administering and monitoring the TOE. The DBW consists of a main window and several sub-windows. The DBW communicates with the Teradata® RDBMS V2R5.0.2 through the Console Subsystem that is a component of the PDE.

2.4.4 Client Node

The Client Node includes Call Level Interface (CLI) software. The CLI interacts directly with the application to accept SQL statements and data going into the Teradata® Server, and return result status and response data. The CLI formats data for the server into a set of parcels and receives a set of parcels in exchange. When the parcels are sent across the LAN, they are preceded by a LAN message header which is constructed by the MTDP when going to the Teradata® Server and read by the MTDP when coming out. The LAN header contains control information for MTDP and the gateway such as the character format of the data in the parcels, the type of action to be taken or the result of an action, the total size of all of the parcels, whether the parcel data is encrypted and the identifier for the session to which the parcels belong. MOSI is the piece of software executing on a Client Node that directly interfaces between the MTDP and LAN software. MOSI maps between the TCP/IP functions used by MTDP and the various versions offered by the different vendors providing the Client Nodes.

2.5 TESTED CONFIGURATION

The TOE in the tested configuration resided on one Database Server. This server consists of a WorldMark 4455 hardware platform running Windows 2000 Advanced Server (Service Pack 3).

3. TOE SECURITY ENVIRONMENT

3.1 INTRODUCTION

This chapter identifies the following:

a) Significant assumptions about the TOE's operational environment,
b) IT related threats to the organization countered by the TOE,
c) Environmental threats requiring controls to provide sufficient protection, and
d) Organizational security policies for the TOE as appropriate.

Using the above listing, this chapter identifies threats (T), organizational security policies (P) and assumptions (A). For assumptions, threats or policies that apply to the environment, the initial character is followed by a period and then an 'E'. For example, T.E.PHYSICAL.

3.2 SECURE USAGE ASSUMPTIONS

The specific conditions listed in the following subsections are assumed to exist in the TOE environment. These assumptions are necessary as a result of practical realities in the development of the TOE security requirements and the essential environmental conditions on the use of the TOE.

3.2.1 TOE Assumptions

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

3.2.2 Physical Assumptions

A.PHYSICAL	The processing resources of the TOE and the underlying system are located within controlled access facilities, which prevents unauthorized physical access.

3.2.3 Configuration Assumptions

A.SYS.CONFIG	The underlying system (operating system and/or secure network services and/or custom software) is installed, configured, and managed in accordance with its secure configuration.
A.MANAGE	There will be one or more individuals who can be trusted not to abuse their privileges in managing the TOE, the underlying system, and the security of the information that they contain.

3.2.4 Connectivity Assumptions

A.PEER	Any other IT components with which the TOE communicates are assumed to be under the same management control and operate under the same security policy.

A.FIREWALL	The network where the TOE is deployed must be protected by a firewall that has been configured to mitigate malicious attacks against the Operating System upon which the TOE operates.

3.3 THREATS TO SECURITY

The threats to security have been categorized based on those against the TOE verses those against the TOE environment.

3.3.1 Threats Against the TOE

T.ABUSE.USER	A user could perform an authorized action that results in an undetected compromise of the DBMS.
T.ACCESS	A user who is not (currently) an authorized user or security administrator accesses the DBMS. This threat includes: Impersonation - a person, who may or may not be an authorized user or security administrator, accesses the database, by impersonating an authorized user or security administrator (including an authorized user impersonating a different user who has different - possibly more privileged - access).
T.ATTACK	An undetected compromise of the database occurs as a result of an attacker (whether an authorized user of the database or not) attempting to perform actions that the individual is not authorized to perform.
T.DATA	An authorized user accesses information contained within a database without the permission of the authorized user who owns or who has responsibility for protecting the data.
T.RESOURCE	An authorized user consumes global database resources, in a way that compromises the ability of other authorized users to access the database.

3.3.2 Threats Against the TOE Environment

T.E.OPERATE	Compromise of the database may occur because of improper configuration, administration, and/or operation of the composite system.
T.E.ACCESS	Because of improper configuration of the underlying operating system, an unauthorized user may obtain access to the system.
T.E.CRASH	Abrupt interruptions to the operation of the TOE may cause security related data, such as database control data and audit data, to be lost or corrupted. Such interruptions may arise from human

	error (see also T.OPERATE) or from failures of software, hardware, power supplies, or storage media.
T.E.NETWORK	Compromise of authentication data may occur if unsecure communication protocols are used to transmit authentication data across the network.
T.E.PHYSICAL	Security-critical parts of the TOE or the underlying operating system and/or network services may be subjected to physical attack, which could compromise security.

3.4 ORGANIZATIONAL SECURITY POLICIES

The following policies apply to the TOE and the intended environment of the TOE.

P.ACCESS — Access to database objects are determined by:

a) The owner of the database object; and
b) The identity of the database subject attempting the access; and
c) The DB object access privileges to the database object held by the database subject; and
d) The resources allocated to the subject.

Note that this policy includes the following:

a) Ownership—database object owners are responsible for their database objects; and
b) Discretionary Access Control—database object owners may grant other authorized users access to or control over their database objects on a discretionary basis.
c) Resources—authorized users are permitted to use only their allocated resources.

P.ACCOUNT — Users are accountable for:

a) Operations on objects as configured by the owner of the object; and
b) Actions configured by the security administrator.

4. SECURITY OBJECTIVES

All of the objectives listed in this section ensure that all of the security threats listed in Chapter 3 have been countered. The security objectives (O) for the Teradata® RDBMS V2R5.0.2 are divided into Security Objectives for the TOE (section 4.1) and Security Objectives for the Environment (section 4.2).

4.1 SECURITY OBJECTIVES FOR THE TOE

The following security objectives are to be satisfied by the TOE.

O.ACCESS — The TOE must provide authorized users and security administrators with the capability of controlling and limiting access, by identified individuals, or grouping of individuals, to the data or resources they own or are responsible for, in accordance with the P.ACCESS security policy. To this end the TOE has the following more specific objectives:

O.ACCESS.OBJECTS — The TOE must prevent the unauthorized or undesired disclosure, entry, modification, or destruction of data and database objects, database views, and database control and audit data.

O.ACCESS.CONTROL — The TOE must allow authorized users who own or are responsible for data to control the access to that data by other authorized users.

O.ACCESS.RESIDUAL — The TOE must prevent unauthorized access to residual data remaining in objects and resources following the use of those objects and resources.

O.ADMIN.TOE — The TOE, where necessary in conjunction with the underlying system, must provide functions to enable a security administrator to effectively manage the TOE and its security functions, ensuring that only the security administrator can access such functionality.

O.AUDIT — The TOE must provide the means of recording security relevant events in sufficient detail to help a security administrator of the TOE to:

a) Detect attempted security violations, or potential misconfiguration of the TOE security features that would leave the database open to compromise; and
b) Hold individual authorized users accountable for any actions they perform that are relevant to the security of the database in accordance with P.ACCOUNT.

O.I&A.TOE The TOE must provide the means of identifying and authenticating users of the TOE.

Applications Note: This security objective explicitly allows identification and authentication of database users to be performed either by the TOE or by the underlying system.

O.PARTIAL.SELF.PROTECTION The TSF will maintain a domain for its own execution that protects itself and its resources from external interference, tampering or unauthorized disclosure, through its own interfaces.

O.RESOURCE The TOE must provide the means of controlling the consumption of database resources by authorized users of the TOE.

4.2 SECURITY OBJECTIVES FOR THE ENVIRONMENT

The following security objectives for the IT environment of the TOE must be satisfied in order for the TOE to fulfill its security objectives.

O.E.ADMIN.ENV The TOE, where necessary in conjunction with the underlying system, must provide functions to enable a security administrator to effectively manage the TOE and its security functions, ensuring that only the security administrator can access such functionality.

O.E.FILES The underlying system must provide access control mechanisms by which all of the database-related files and directories (including executables and run-time libraries) may be protected from unauthorized access.

O.E.FIREWALL The network where the TOE is deployed must be protected by a firewall that has been configured to mitigate malicious attacks against the Operating System upon which the TOE operates.

O.E.SEP The underlying operating system must provide the means to isolate the TOE Security Functions (TSF) and assure that TSF components cannot be tampered with. The TSF components are 1) the devices used to store the database and 2) the TOE processes managing the database.

The following non-IT security objectives are to be satisfied by procedural and other measures taken within the TOE environment.

O.E.AUDITLOG The security administrator of the database must ensure that audit facilities are used and managed effectively. These procedures shall apply to the database audit trail and/or the audit trail for the underlying operating system and/or secure network services. In particular:

a) Appropriate action must be taken to ensure continued audit logging, e.g., by regular archiving of logs before audit trail exhaustion to ensure sufficient free space.
b) Audit logs must be inspected on a regular basis and appropriate action should be taken on the detection of breaches of security, or events that are likely to lead to a breach in the future.
c) The system clocks must be protected from unauthorized modification (so that the integrity of the audit timestamps is not compromised).

O.E.AUTHDATA Those responsible for the TOE must ensure that the authentication data for each user account for the TOE as well as the underlying system is held securely and not disclosed to persons not authorized to use that account. In particular:

a) The media on which the authentication data for the underlying operating system and/or secure network services is stored shall not be physically removable from the underlying platform by unauthorized users;
b) Authorized users shall not disclose their passwords to other individuals;
c) Passwords generated by the security administrator shall be distributed in a secure manner.

O.E.INSTALL Those responsible for the TOE must ensure that:

a) The TOE is delivered, installed, managed, and operated in accordance with the operational documentation of the TOE, and
b) The underlying system is installed and operated in accordance with its operational documentation

O.E.PHYSICAL Those responsible for the TOE must ensure that those parts of the TOE that are critical to the security policy are protected from physical attack.

O.E.QUOTA The security administrator of the database must ensure that each authorized user of the TOE is configured with appropriate quotas that are:

a) Sufficiently permissive to allow the user to perform the operations for which the user has access;
b) Sufficiently restrictive that the user cannot abuse the access and thereby monopolize resources.

O.E.RECOVERY Those responsible for the TOE must ensure that procedures and/or mechanisms are in place to ensure that, after system failure or

other discontinuity, recovery without protection (i.e. security) compromise is obtained.

O.E.TRUST — Those responsible for the TOE must ensure that only highly trusted users have the privilege that allows them to:

a) Set or alter the audit trail configuration for the database;
b) Alter or delete any audit record in the database audit trail;
c) Create any user account or modify any user security attributes;
d) Authorize use of administrative privileges.

O.E.MEDIA — Those responsible for the TOE must ensure that the confidentiality, integrity and availability of data held on storage media is adequately protected. In particular:

a) The on-line and off-line storage media on which database and security related data (such as operating system backups, database backups and transaction logs, and audit trails) must not be physically removable from the underlying platform by unauthorized users.
b) The on-line and off-line storage media must be properly stored and maintained, and routinely checked to ensure the integrity and availability of the security related data.
c) The media on which database-related files (including database files, export files, redo log files, control files, trace files, and dump files) have been stored shall be purged prior to being re-used for any non-database purpose.

4.3 SECURITY OBJECTIVES MAPPING

The following table provides a mapping with rationale to identify the security objectives that address the stated threats and organizational security policies.

Table 4-1 TOE Threat and Policy Mapping to Objectives

Threats & Policies	Security Objectives	Rationale
T.ABUSE.USER	O.ACCESS.CONTROL, O.ADMIN.TOE, O.AUDIT, O.I&A.TOE	O.AUDIT ensures the TOE has the means of recording security relevant events that could be indicative of abuse of privilege by an authorized user of the database (whether intentional or otherwise). O.I&A.TOE provides support by reliably identifying the user responsible for particular events, thus ensuring that the user can be held accountable for actions for which he or she is responsible. O.ACCESS.CONTROL and O.ADMIN.TOE provide support by controlling access to audit configuration data that only highly trusted individuals must be allowed to view and modify.

Threats & Policies	Security Objectives	Rationale
T.ACCESS	O.ACCESS.CONTROL, O.ADMIN.TOE, O.I&A.TOE, O.RESOURCE	O.I&A.TOE ensures the TOE with or without an underlying operating system has the means to protect global data and resources of the database from unauthorized personnel by identifying and authentication users of the TOE. O.ACCESS.CONTROL, O.ADMIN.TOE and O.RESOURCES provide support by controlling access to database control data and administrative functionality.
T.ATTACK	O.ACCESS.CONTROL, O.ADMIN.TOE, O.AUDIT, O.I&A.TOE	O.AUDIT ensures the TOE has a means of recording security relevant events that could be indicative of an attack at defeating TSF. O.I&A.TOE provides support by identifying the user responsible for particular events, where the attacker is an authorized user. O.ACCESS.CONTROL and O.ADMIN.TOE together provide support by controlling access to audit configuration data that only highly trusted individuals must be allowed to view and modify.
T.DATA	O.ACCESS.OBJECTS, O.ACCESS.CONTROL, O.ACCESS.RESIDUAL, O.ADMIN.TOE, O.I&A.TOE	O.ACCESS.OBJECTS ensures access is controlled to information contained within specific database objects. O.ACCESS.RESIDUAL ensures access is prevented to residual information stored in a database table. 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.
T.RESOURCE	O.ACCESS.CONTROL, O.ADMIN.TOE, O.I&A.TOE,	O.I&A.TOE provides support by providing the means of identifying the user attempting to use resources. 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 resource utilization controls.
P.ACCESS	O.ACCESS.OBJECTS, O.RESOURCE, O.PARTIAL.SELF.PROTECTION	O.ACCESS.OBJECTS ensures that the subjects using the TOE are able to control access to the objects that they own or for which they are responsible. O.RESOURCE ensures that the TOE is able to control the consumption of resources. The TOE therefore provides partial self-protection through its database-specific access controls (O.PARTIAL.SELF.PROTECTION).
P.ACCOUNT	O.ACCESS, O.AUDIT	O.AUDIT ensures that the subjects using the TOE are accountable for their actions by recording details of attempted security violations and other actions that have been configured for auditing. O.ACCESS ensures that the accounting data is protected.

Table 4-2 TOE Environment Threat, Policy, and Assumption Mapping to Objectives

Assumptions, Threats & Policies	Security Objectives	Rationale
A.TOE.CONFIG	O.E.INSTALL	O.E.INSTALL ensures that the TOE software is configured in a manner that establishes a secure baseline state of operation.
A.PHYSICAL	O.E.PHYSICAL	O.E.PHYSICAL addresses the need for physical protection of the TOE in the environment that it resides.
A.SYS.CONFIG	O.E.INSTALL	O.E.INSTALL ensures that the underlying operating system upon which the TOE is installed has been configured in a manner that establishes a secure baseline state of operation.
A.MANAGE	O.E.INSTALL, O.E.AUDITLOG, O.E.RECOVERY, O.E.QUOTA, O.E.TRUST, O.E.AUTHDATA, O.E.MEDIA, O.E.ADMIN.ENV, O.E.SEP, O.E.FILES	The combined listing of security objectives address the assumption in that administration and management of the TOE functions are performed by users that can be trusted given their assigned privileges.
A.PEER	O.E.ADMIN.ENV	O.E.ADMIN.ENV ensures that IT components that communicate with the TOE are deployed in a manner that they provide secure administrative access to the TOE.
A.FIREWALL	O.E.FIREWALL	O.E.FIREWALL addresses the assumption A.FIREWALL by restatement.
T.E.NETWORK	O.E.AUTHDATA	Provided by O.E.AUTHDATA since the network may be used to transport authentication data.
T.E.OPERATE	O.E.INSTALL	O.E.INSTALL ensures that the TOE and its underlying platform are correctly installed, managed and operated.
T.E.ACCESS	O.E.PHYSICAL, O.E.ADMIN_ENV, O.E.FILES	O.E.PHYSICAL counters physical attacks that represent access threats, O.E.ADMIN_ENV ensures that the security administrator has management control of the security policies of components that provide access, and O.E.FILES calls on the underlying file system to support TOE access controls with its own access control mechanisms.
T.E.CRASH	O.E.RECOVERY, O.E.MEDIA	O.E.MEDIA and O.E.RECOVERY ensure that suitable recovery mechanisms are in place to recover from a crash and that the media used during the crash recovery is able to maintain the confidentially, integrity and availability of the TOE.
T.E.PHYSICAL	O.E.PHYSICAL	O.E.PHYSICAL protects critical parts of the TOE from physical attack.

Assumptions, Threats & Policies	Security Objectives	Rationale
P.ACCESS	O.ACCESS, O.E.AUTHDATA, O.E.FILES, O.E.SEP	The TOE relies on the underlying OS's user accounts and file structure (O.E.AUTHDATA and O.E.FILES) and relies on the underlying OS to ensure isolation of the TSF (O.E.SEP). It builds upon O.E.AUTHDATA to provide access control mechanisms that pertain directly to its own database objects (O.ACCESS). Therefore, even as a software only TOE, it does provide partial self-protection through its database specific access controls.
P.ACCOUNT	O.AUDIT, O.E.AUDITLOG	The TOE is required, via O.AUDIT, to provide auditing functions that ensure accountability for accesses of the database objects. The underlying OS provides additional functionality, through O.E.AUDITLOG, by auditing basic OS services that pertain to the database software.

5. SECURITY REQUIREMENTS

This section identifies the security functional requirements for the TOE and its' IT environment. In addition, this section also presents the security assurance requirements for the TOE. The operations performed on the security functional and assurance requirements contained in this section adhere to the following conventions:

- Iteration: Allows a component to be used more than once with varying operations. In the ST, a number in parenthesis appended to a component indicates iteration. For example, FMT_MOF.1 Management of security functions behaviour (1) and FMT_MOF.1 Management of security functions behaviour (2) indicate that the ST includes two iterations of the FMT_MOF.1 component.

- Assignment: Allows the specification of an identified parameter. Assignments are indicated using italicized text and are surrounded by brackets (e.g., [*assignment*]).

- Selection: Allows the specification of one or more elements from a list. Selections are indicated using bold italicized text and are surrounded by brackets (e.g., [***selection***]).

- Refinement: Allows the addition of details. Refinements are indicated using bold text for additions to the requirements (e.g., **refinement**). In addition, refinements based upon CCIMB interpretations are indicated in red italicized text for additions, and strikethrough red italicized text for deletions (e.g., *text added* ~~*text removed*~~).

5.1 TOE SECURITY FUNCTIONAL REQUIREMENTS

The following table provides a summary of the security functional requirements implemented by the TOE.

Table 5-1 TOE Security Functional Requirements

Security Functional Class	Security Functional Component
Security Audit (FAU)	FAU_GEN.1 Audit data generation
	FAU_GEN.2 User identity association
	FAU_SAR.1 Audit review
	FAU_SAR.3 Selectable audit review
	FAU_SEL.1 Selective audit
	FAU_STG_EXP.1 Protected audit trail storage
User Data Protection (FDP)	FDP_ACC.1 Subset access control
	FDP_ACF.1 Security attribute based access control
	FDP_RIP.1 Subset residual information protection
Identification and Authentication (FIA)	FIA_AFL.1 Authentication failure handling
	FIA_ATD.1 User attribute definition
	FIA_SOS.1 Verification of secrets
	FIA_UAU.1 Timing of authentication
	FIA_UID.1 Timing of identification
	FIA_USB.1 User-subject binding

Security Functional Class	Security Functional Component
Security Management (FMT)	FMT_MOF.1 Management of security functions behaviour (1)
	FMT_MOF.1 Management of security functions behaviour (2)
	FMT_MSA.1 Management of security attributes (1)
	FMT_MSA.1 Management of security attributes (2)
	FMT_MSA.1 Management of security attributes (3)
	FMT_MSA.1 Management of security attributes (4)
	FMT_MSA.1 Management of security attributes (5)
	FMT_MSA.3 Static attribute initialisation
	FMT_MTD.1 Management of TSF data
	FMT_REV.1 Revocation
	FMT_SMF.1 Specification of management functions
	FMT_SMR.1 Security roles
Protection of the TSF (FPT)	FPT_RVM.1 Non-bypassability of the TSP
	FPT_SEP_EXP.1 TSF domain separation
Resource Utilisation (FRU)	FRU_RSA.1 Maximum quotas
TOE Access (FTA)	FTA_TSE.1 TOE session establishment

The following subsections present the security functional requirements for the TOE.

5.1.1 Security Audit (FAU)

5.1.1.1 FAU_GEN.1 Audit data generation

Hierarchical to: No other components.

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

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

b) All auditable events for the [***not specified***] level of audit; and

c) [*Events specified in Table 5-2*].

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

a) Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event; and

b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [*other audit relevant information, as provided under "Additional Data" in Table 5-2*]

Table 5-2 Auditable Events

Component	Event	Additional Data
FAU_SAR.1	Reading of information from the database audit records	None
FAU_SEL.1	All modifications to the database audit configuration that occur while the database audit collection functions are operating	Modified configuration element
FDP_ACF.1	All requests to perform an operation on a database object covered by the SFP	The assigned user privilege
FIA_UAU.1	Use of the user authentication mechanism	None
FIA_UID.1	Use of the user identification mechanism	None
FIA_USB.1	Success or failure of binding user security attributes to a database subject (e.g., success and failure to create a database subject)	None
FMT_MOF.1	Modifications in the behaviour of the functions of the TSF	Change of threshold for unsuccessful authentication attempts or actions to be taken in the event of an authentication failure
FMT_MSA.1	Modifications of the values of database security attributes	Modification, deletion or addition of database security attributes
FMT_MTD.1	Modifications to the values of TSF data	None
FMT_REV.1	Attempts to revoke database security attributes	None
FMT_SMR.1	Modifications to the group of users that are part of a role	None
FTA_TSE.1	Unsuccessful attempts at establishment of a user session	None

Dependencies: FPT_STM.1 Reliable time stamps

5.1.1.2 FAU_GEN.2 User identity association

Hierarchical to: No other components.

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

Dependencies: FAU_GEN.1 Audit data generation

FIA_UID.1 Timing of identification

5.1.1.3 FAU_SAR.1 Audit review

Hierarchical to: No other components.

FAU_SAR.1.1 The TSF shall provide [*the security administrator*] with the capability to read [*all audit information*] from the audit records.

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

Dependencies: FAU_GEN.1 Audit data generation

5.1.1.4 FAU_SAR.3 Selectable audit review

Hierarchical to: No other components.

FAU_SAR.3.1 The TSF shall provide the ability to perform [***searches, sorting***] of audit data based on [*all attributes contained within the audit records*].

Dependencies: FAU_SAR.1 Audit review

5.1.1.5 FAU_SEL.1 Selective audit

Hierarchical to: No other components.

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

a) [***event type***]

b) [*subject identity, object identity*].

Dependencies: FAU_GEN.1 Audit data generation

FMT_MTD.1 Management of TSF data

5.1.1.6 FAU_STG_EXP.1 Protected audit trail storage

Hierarchical to: No other components.

FAU_STG_EXP.1.1 For actions within the TOE Scope of Control, the TSF shall protect the stored audit records from unauthorised deletion.

FAU_STG_EXP.1.2 When attempts to modify audit records occur within the TOE Scope of Control, the TSF shall be able to [***prevent***] unauthorised modifications to the audit records in the audit trail.

Dependencies: FAU_GEN.1 Audit data generation

5.1.2 User Data Protection (FDP)

5.1.2.1 FDP_ACC.1 Subset access control

Hierarchical to: No other components.

FDP_ACC.1.1 The TSF shall enforce the [*Database Object Access Control SFP*] on [*user identity, databases, tables, views, macros, stored procedures, and all permitted operations on database objects by database subjects covered by this SFP*].

Dependencies: FDP_ACF.1 Security attribute based access control

5.1.2.2 FDP_ACF.1 Security attribute based access control

Hierarchical to: No other components.

Interp Note: *The following element is changed as a result of Interpretation 103.*

FDP_ACF.1.1 The TSF shall enforce the [*Database Object Access Control SFP*] to objects based on *the following:* [*elements contained in Table 5-3*].

Table 5-3 Access Control Attributes

Subjects and Objects	Security Attributes
Users interactively querying the database	User identity and role
Queries sent from an application program	User identity associated with the application
Databases, tables, views, macros, stored procedures	Administrative or database object access privileges

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

a) *If the authorized user associated with the subject is the owner of the object, then the requested access is allowed; or*

b) *If the subject has the object access privilege for the requested access to the object, then the requested access is allowed; or*

c) *If the subject is the member of a role which has the object access privilege for the requested access to the object, then the requested access is allowed; or*

d) *Otherwise, the access is denied*].

FDP_ACF.1.3 The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: [*no additional rules*].

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the [*no additional rules*].

Dependencies: FDP_ACC.1 Subset access control

FMT_MSA.3 Static attribute initialization

5.1.2.3 FDP_RIP.1 Subset residual information protection

Hierarchical to: No other components.

FDP_RIP.1.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon the [***allocation of the resource to***] the following objects: [*database tables*].

Dependencies: No dependencies

5.1.3 Identification and Authentication (FIA)

5.1.3.1 FIA_AFL.1 Authentication failure handling

Hierarchical to: No other components.

FIA_AFL.1.1 The TSF shall detect when [*three*] unsuccessful authentication attempts occur related to [*the unsuccessful authentication attempts since the last successful authentication for the indicated user identity*].

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been met or surpassed, the TSF shall [*disable the account until unlocked by the security administrator*].

Dependencies: FIA_UAU.1 Timing of authentication

5.1.3.2 FIA_ATD.1 User attribute definition

Hierarchical to: No other components.

FIA_ATD.1.1 The TSF shall maintain the following list of security attributes belonging to individual users: [*user identity, role, password, profile*].

Dependencies: No dependencies

5.1.3.3 FIA_SOS.1 Verification of secrets

Hierarchical to: No other components.

FIA_SOS.1.1 The TSF shall provide a mechanism to verify that secrets meet [*a minimum requirement of 8 characters in length*].

Dependencies: No dependencies

5.1.3.4 FIA_UAU.1 Timing of authentication

Hierarchical to: No other components.

FIA_UAU.1.1 The TSF shall allow [*establishment of a virtual circuit, receipt of error messages upon authentication failure*] on behalf of the user to be performed before the user is authenticated.

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

Dependencies: FIA_UID.1 Timing of identification

5.1.3.5 FIA_UID.1 Timing of identification

Hierarchical to: No other components.

FIA_UID.1.1 The TSF shall allow [*establishment of a virtual circuit for the purpose of transferring authentication information, receipt of error messages upon identification failure*] on behalf of the user to be performed before the user is identified.

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

Dependencies: No dependencies

5.1.3.6 FIA_USB.1 User-subject binding

Hierarchical to: No other components.

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

Dependencies: FIA_ATD.1 User attribute definition

5.1.4 Security Management (FMT)

5.1.4.1 FMT_MOF.1 Management of security functions behaviour (1)

Hierarchical to: No other components.

FMT_MOF.1.1(1) The TSF shall restrict the ability to [***enable, modify the behaviour of***] the functions [*threshold for unsuccessful authentication attempts*] to [*the security administrator*].

Interp Note: *This list of dependencies is changed as a result of Interpretation 065.*

Dependencies: *FMT_SMF.1 Specification of management functions*

FMT_SMR.1 Security roles

5.1.4.2 FMT_MOF.1 Management of security functions behaviour (2)

Hierarchical to: No other components.

FMT_MOF.1.1(2) The TSF shall restrict the ability to [***determine the behaviour of***] the functions [*actions to be taken in the event of an authentication failure*] to [*the security administrator*].

Interp Note: *This list of dependencies is changed as a result of Interpretation 065.*

Dependencies: *FMT_SMF.1 Specification of management functions*

FMT_SMR.1 Security roles

5.1.4.3 FMT_MSA.1 Management of security attributes (1)

Hierarchical to: No other components.

FMT_MSA.1.1(1) The TSF shall enforce the [*Database Object Access Control SFP*] to restrict the ability to [***modify, delete,*** [*or add*]] the security attributes [*database object access privileges*] to [*the security administrator or authorised user*].

Interp Note: *This list of dependencies is changed as a result of Interpretation 065.*

Dependencies: [FDP_ACC.1 Subset access control or

FDP_IFC.1 Subset information flow control]

FMT_SMF.1 Specification of management functions

FMT_SMR.1 Security roles

5.1.4.4 FMT_MSA.1 Management of security attributes (2)

Hierarchical to: No other components.

FMT_MSA.1.1(2) The TSF shall enforce the [*Database Object Access Control SFP*] to restrict the ability to [***modify, delete,*** [*or add*]] the security attributes [*database roles*] to [*the security administrator*].

Interp Note: *This list of dependencies is changed as a result of Interpretation 065.*

Dependencies: [FDP_ACC.1 Subset access control or

FDP_IFC.1 Subset information flow control]

FMT_SMF.1 Specification of management functions

FMT_SMR.1 Security roles

5.1.4.5 FMT_MSA.1 Management of security attributes (3)

Hierarchical to: No other components.

FMT_MSA.1.1(3) The TSF shall enforce the [*Database Object Access Control SFP*] to restrict the ability to [***modify, delete,*** [*or add*]] the security attributes [*user identities*] to [*the security administrator*].

Interp Note: *This list of dependencies is changed as a result of Interpretation 065.*

Dependencies: [FDP_ACC.1 Subset access control or

FDP_IFC.1 Subset information flow control]

FMT_SMF.1 Specification of management functions

FMT_SMR.1 Security roles

5.1.4.6 FMT_MSA.1 Management of security attributes (4)

Hierarchical to: No other components.

FMT_MSA.1.1(4) The TSF shall enforce the [*Database Object Access Control SFP*] to restrict the ability to [***change_default***] the security attributes [*user assigned role*] to [*the security administrator*].

Interp Note: *This list of dependencies is changed as a result of Interpretation 065.*

Dependencies: [FDP_ACC.1 Subset access control or

FDP_IFC.1 Subset information flow control]

FMT_SMF.1 Specification of management functions

FMT_SMR.1 Security roles

5.1.4.7 FMT_MSA.1 Management of security attributes (5)

Hierarchical to: No other components.

FMT_MSA.1.1(5) The TSF shall enforce the [*Database Object Access Control SFP*] to restrict the ability to [***modify, delete,*** [*or add*]] the security attributes [*maximum quotas*] to [*the security administrator*].

Interp Note: *This list of dependencies is changed as a result of Interpretation 065.*

Dependencies: [FDP_ACC.1 Subset access control or

FDP_IFC.1 Subset information flow control]

FMT_SMF.1 Specification of management functions

FMT_SMR.1 Security roles

5.1.4.8 FMT_MSA.3 Static attribute initialisation

Hierarchical to: No other components.

FMT_MSA.3.1 The TSF shall enforce the [*Database Object Access Control SFP*] to provide [***restrictive***] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow the [*no identified roles*] to specify alternative initial values to override the default values when an object or information is created.

Dependencies: FMT_MSA.1 Management of security attributes

FMT_SMR.1 Security roles

5.1.4.9 FMT_MTD.1 Management of TSF data

Hierarchical to: No other components.

FMT_MTD.1.1 The TSF shall restrict the ability to [***modify, delete,*** [*or add*]] the [*user IDs, authentication data and roles*] to [*the security administrator*].

Interp Note: This list of dependencies is changed as a result of Interpretation 065.

Dependencies: *FMT_SMF.1 Specification of management functions*

FMT_SMR.1 Security roles

5.1.4.10 FMT_REV.1 Revocation

Hierarchical to: No other components.

FMT_REV.1.1 The TSF shall restrict the ability to revoke security attributes associated with the [***users, objects***] within the TSC to [*authorised 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)*].

FMT_REV.1.2 The TSF shall enforce the rules [*revocation of database object access privileges shall effect all subsequent attempts to establish access to the database object*].

Dependencies: FMT_SMR.1 Security roles

Interp note: The following component is created as a result of Interpretation 065.

5.1.4.11 *FMT_SMF.1 Specification of management functions*

Hierarchical to: No other components.

FMT_SMF.1.1 *The TSF shall be capable of performing the following security management functions [management of users and roles, management of*

functions in the TSF, management of security attributes, static attribute initialisation].

Dependencies: *No dependencies*

5.1.4.12 FMT_SMR.1 Security roles

Hierarchical to: No other components.

FMT_SMR.1.1 The TSF shall maintain the roles [*security administrator and authorised user*].

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

Dependencies: FIA_UID.1 Timing of identification

5.1.5 Protection of the TSF (FPT)

5.1.5.1 FPT_RVM.1 Non-bypassability of the TSP

Hierarchical to: No other components.

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.

Dependencies: No dependencies

5.1.5.2 FPT_SEP_EXP.1 TSF domain separation

Hierarchical to: No other components.

FPT_SEP_EXP.1.1 The TSF shall maintain a security domain that protects it from interference and tampering by untrusted subjects initiating actions through its own TSFI.

Dependencies: No dependencies

5.1.6 Resource Utilisation (FRU)

5.1.6.1 FRU_RSA.1 Maximum quotas

Hierarchical to: No other components.

FRU_RSA.1.1 The TSF shall enforce maximum quotas of the following resources: [*CPU processing time for each job session, logical data blocks used for a specified job, database storage allocations*] that [***individual user***] can use [***simultaneously***].

Dependencies: No dependencies

5.1.7 TOE Access (FTA)

5.1.7.1 FTA_TSE.1 TOE session establishment

Hierarchical to: No other components.

FTA_TSE.1.1 The TSF shall be able to deny session establishment based on [*user identity*].

Dependencies: No dependencies

5.2 IT ENVIRONMENT SECURITY FUNCTIONAL REQUIREMENTS

This section identifies the security functional requirements that have been levied onto the IT environment that must be implemented in order for the TOE to enforce its' stated functional claims.

The following table provides a summary of the security functional requirements that must be implemented by the IT environment.

Table 5-4 IT Environment Security Functional Requirements

Security Functional Class	Security Functional Component
Protection of the TSF (FPT)	FPT_SEP_ENV_EXP.1 Domain separation
	FPT_STM.1 Reliable time stamps

The following subsections present the security functional requirements for the IT environment.

5.2.1 Protection of the TSF (FPT)

5.2.1.1 FPT_SEP_ENV_EXP.1 Domain separation

Hierarchical to: No other components.

FPT_SEP_ENV_EXP.1.1 The IT Environment shall provide hardware that provides virtual memory management and at least two execution rings for executing software.

Dependencies: No dependencies

5.2.1.2 FPT_STM.1 Reliable time stamps

Hierarchical to: No other components.

FPT_STM.1.1 The **IT Environment** shall be able to provide reliable time-stamps for use **by the TOE**.

Dependencies: No dependencies

5.3 TOE SECURITY ASSURANCE REQUIREMENTS

This section identifies the security assurance requirements that are met by the TOE. These assurance requirements conform to the CC Part 3 requirements for EAL2 and are identified in the following table.

Table 5-5 TOE Security Assurance Requirements

Security Assurance Class	Security Assurance Component
Configuration Management (ACM)	ACM_CAP.2 Configuration items
Delivery and Operation (ADO)	ADO_DEL.1 Delivery procedures
	ADO_IGS.1 Installation, generation, and start-up procedures
Development (ADV)	ADV_FSP.1 Informal functional specification
	ADV_HLD.1 Descriptive high-level design
	ADV_RCR.1 Informal correspondence demonstration
Guidance Documents (AGD)	AGD_ADM.1 Administrator guidance
	AGD_USR.1 User guidance
Tests (ATE)	ATE_COV.1 Evidence of coverage
	ATE_FUN.1 Functional testing
	ATE_IND.2 Independent testing – sample
Vulnerability Assessment (AVA)	AVA_SOF.1 Strength of TOE security function evaluation
	AVA_VLA.1 Developer vulnerability analysis

The following subsections present the security assurance requirements for the TOE.

5.3.1 Configuration management (ACM)

5.3.1.1 Configuration items (ACM_CAP.2)

ACM_CAP.2.1D The developer shall provide a reference for the TOE.

ACM_CAP.2.2D The developer shall use a CM system.

ACM_CAP.2.3D The developer shall provide CM documentation.

ACM_CAP.2.1C The reference for the TOE shall be unique to each version of the TOE.

ACM_CAP.2.2C The TOE shall be labeled with its reference.

ACM_CAP.2.3C The CM documentation shall include a configuration list.

Interp Note: *The following element is added as a result of Interpretation 003.*

The configuration list shall uniquely identify all configuration items that comprise the TOE.

ACM_CAP.2.4C The configuration list shall describe the configuration items that comprise the TOE.

ACM_CAP.2.5C The CM documentation shall describe the method used to uniquely identify the configuration items.

ACM_CAP.2.6C The CM system shall uniquely identify all configuration items.

ACM_CAP.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.2 Delivery and operation (ADO)

5.3.2.1 Delivery procedures (ADO_DEL.1)

ADO_DEL.1.1D The developer shall document procedures for delivery of the TOE or parts of it to the user.

ADO_DEL.1.2D The developer shall use the delivery procedures.

ADO_DEL.1.1C The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to a user's site.

ADO_DEL.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.2.2 Installation, generation, and start-up procedures (ADO_IGS.1)

ADO_IGS.1.1D The developer shall document procedures necessary for the secure installation, generation, and start-up of the TOE.

Interp Note: *The following element has changed as a result of Interpretation 051.*

ADO_IGS.1.1C The *installation, generation and start-up* documentation shall describe *all* the steps necessary for secure installation, generation, and start-up of the TOE.

ADO_IGS.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADO_IGS.1.2E The evaluator shall determine that the installation, generation, and start up procedures result in a secure configuration.

5.3.3 Development (ADV)

5.3.3.1 Informal functional specification (ADV_FSP.1)

ADV_FSP.1.1D The developer shall provide a functional specification.

ADV_FSP.1.1C The functional specification shall describe the TSF and its external interfaces using an informal style.

ADV_FSP.1.2C The functional specification shall be internally consistent.

ADV_FSP.1.3C The functional specification shall describe the purpose and method of use of all external TSF interfaces, providing details of effects, exceptions and error messages, as appropriate.

ADV_FSP.1.4C The functional specification shall completely represent the TSF.

ADV_FSP.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_FSP.1.2E The evaluator shall determine that the functional specification is an accurate and complete instantiation of the TOE security functional requirements.

5.3.3.2 Descriptive high-level design (ADV_HLD.1)

ADV_HLD.1.1D The developer shall provide the high-level design of the TSF.

ADV_HLD.1.1C The presentation of the high-level design shall be informal.

ADV_HLD.1.2C The high-level design shall be internally consistent.

ADV_HLD.1.3C The high-level design shall describe the structure of the TSF in terms of subsystems.

ADV_HLD.1.4C The high-level design shall describe the security functionality provided by each subsystem of the TSF.

ADV_HLD.1.5C The high-level design shall identify any underlying hardware, firmware, and/or software required by the TSF with a presentation of the functions provided by the supporting protection mechanisms implemented in that hardware, firmware, or software.

ADV_HLD.1.6C The high-level design shall identify all interfaces to the subsystems of the TSF.

ADV_HLD.1.7C The high-level design shall identify which of the interfaces to the subsystem of the TSF are externally visible.

ADV_HLD.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_HLD.1.2E The evaluator shall determine that the high-level design is an accurate and complete instantiation of the TOE security functional requirements.

5.3.3.3 Informal correspondence demonstration (ADV_RCR.1)

ADV_RCR.1.1D The developer shall provide an analysis of correspondence between all adjacent pairs of TSF representations that are provided.

ADV_RCR.1.1C For each adjacent pair of provided TSF representations, the analysis shall demonstrate that all relevant security functionality of the more abstract TSF representation is correctly and completely refined in the less abstract TSF representation.

ADV_RCR.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.4 Guidance documents (AGD)

5.3.4.1 Administrator guidance (AGD_ADM.1)

AGD_ADM.1.1D The developer shall provide administrator guidance addressed to system administrative personnel.

AGD_ADM.1.1C The administrator guidance shall describe the administrative functions and interfaces available to the administrator of the TOE.

AGD_ADM.1.2C The administrator guidance shall describe how to administer the TOE in a secure manner.

AGD_ADM.1.3C The administrator guidance shall contain warnings about functions and privileges that should be controlled in a secure processing environment.

AGD_ADM.1.4C The administrator guidance shall describe all assumptions regarding user behavior that are relevant to secure operation of the TOE.

AGD_ADM.1.5C The administrator guidance shall describe all security parameters under the control of the administrator, indicating secure values as appropriate.

AGD_ADM.1.6C The administrator guidance shall describe each type of security relevant event relative to the administrative functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD_ADM.1.7C The administrator guidance shall be consistent with all other documentation supplied for evaluation.

AGD_ADM.1.7C The administrator guidance shall be consistent with all other documentation supplied for evaluation.

AGD_ADM.1.8C The administrator guidance shall describe all security requirements for the IT environment that are relevant to the administrator.

AGD_ADM.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.4.2 User guidance (AGD_USR.1)

AGD_USR.1.1D The developer shall provide user guidance.

AGD_USR.1.1C The user guidance shall describe the functions and interfaces available to the non-administrative users of the TOE.

AGD_USR.1.2C The user guidance shall describe the use of user-accessible security functions provided by the TOE.

AGD_USR.1.3C The user guidance shall contain warnings about user-accessible functions and privileges that should be controlled in a secure processing environment.

AGD_USR.1.4C The user guidance shall clearly present all user responsibilities necessary for secure operation of the TOE, including those related to assumptions regarding user behavior found in the statement of TOE security environment.

AGD_USR.1.5C The user guidance shall be consistent with all other documentation supplied for evaluation.

AGD_USR.1.6C The user guidance shall describe all security requirements for the IT environment that are relevant to the user.

AGD_USR.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.5 Tests (ATE)

5.3.5.1 Evidence of coverage (ATE_COV.1)

ATE_COV.1.1D The developer shall provide evidence of the test coverage.

ATE_COV.1.1C The evidence of the test coverage shall show the correspondence between the tests identified in the test documentation and the TSF as described in the functional specification.

ATE_COV.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.5.2 Functional testing (ATE_FUN.1)

ATE_FUN.1.1D The developer shall test the TSF and document the results.

ATE_FUN.1.2D The developer shall provide test documentation.

ATE_FUN.1.1C The test documentation shall consist of test plans, test procedure descriptions, expected test results and actual test results.

ATE_FUN.1.2C The test plans shall identify the security functions to be tested and describe the goal of the tests to be performed.

ATE_FUN.1.3C The test procedure descriptions shall identify the tests to be performed and describe the scenarios for testing each security function. These scenarios shall include any ordering dependencies on the results of other tests.

ATE_FUN.1.4C The expected test results shall show the anticipated outputs from a successful execution of the tests.

ATE_FUN.1.5C The test results from the developer execution of the tests shall demonstrate that each tested security function behaved as specified.

ATE_FUN.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.5.3 Independent testing - sample (ATE_IND.2)

ATE_IND.2.1D The developer shall provide the TOE for testing.

ATE_IND.2.1C The TOE shall be suitable for testing.

ATE_IND.2.2C The developer shall provide an equivalent set of resources to those that were used in the developer's functional testing of the TSF.

ATE_IND.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_IND.2.2E The evaluator shall test a subset of the TSF as appropriate to confirm that the TOE operates as specified.

ATE_IND.2.3E The evaluator shall execute a sample of tests in the test documentation to verify the developer test results.

5.3.6 Vulnerability assessment (AVA)

5.3.6.1 Strength of TOE security function evaluation (AVA_SOF.1)

AVA_SOF.1.1D The developer shall perform a strength of TOE security function analysis for each mechanism identified in the ST as having a strength of TOE security function claim.

AVA_SOF.1.1C For each mechanism with a strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the minimum strength level of SOF-basic.

AVA_SOF.1.2C For each mechanism with a specific strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the specific strength of function metric of SOF-basic.

AVA_SOF.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_SOF.1.2E The evaluator shall confirm that the strength claims are correct.

5.3.6.2 Developer vulnerability analysis (AVA_VLA.1)

Interp Note: *The following two elements are changed as a result of Interpretation 051.*

AVA_VLA.1.1D ~~*The developer shall perform and document an analysis of the TOE deliverables searching for obvious ways in which a user can violate the TSP.*~~

AVA_VLA.1.2D ~~*The developer shall document the disposition of obvious vulnerabilities.*~~

The developer shall perform a vulnerability analysis.

The developer shall provide vulnerability analysis documentation.

Interp Note: *The following element is replaced by three as a result of Interpretation 051.*

AVA_VLA.1.1C ~~*The documentation shall show, for all identified vulnerabilities, that the vulnerability cannot be exploited in the intended environment for the TOE.*~~

The vulnerability analysis documentation shall describe the analysis of the TOE deliverables performed to search for obvious ways in which a user can violate the TSP.

The vulnerability analysis documentation shall describe the disposition of obvious vulnerabilities.

The vulnerability analysis documentation shall show, for all identified vulnerabilities, that the vulnerability cannot be exploited in the intended environment for the TOE.

AVA_VLA.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_VLA.1.2E The evaluator shall conduct penetration testing, building on the developer vulnerability analysis, to ensure obvious vulnerabilities have been addressed.

6. TOE SUMMARY SPECIFICATION

This chapter describes the high-level specification of each TOE Security Function (TSF) that contributes to satisfaction of the SFRs presented in Chapter 5. It also details the Assurance Measures applied to ensure the correct implementation of the SFRs.

6.1 TOE SECURITY FUNCTIONS

Each of the following subsections describes a security function of the Teradata® RDBMS V2R5.0.2. For each security function, details are provided that substantiate how the Teradata® RDBMS V2R5.0.2 meets the security function, and ensures that no function can be subverted or bypassed without being traced. At the end of each subsection, the SFRs that are satisfied by the TSF are listed, and when it provides added clarity, short additional details specific to the TSF are provided.

6.1.1 TOE Access

Users are identified using a user id. The user id is a unique key in the user table, and so cannot be assigned to more than one user. The data dictionary for the user account table includes columns for RoleName and ProfileName.

The TOE access function implements satisfies the following security requirements:

- FTA_TSE.1 TOE session establishment

6.1.2 Identification and Authentication

The session controller module is primarily responsible for user identification and authentication, including access logging for each session connect and disconnect. It uses one-way encryption algorithm to encrypt the password string and compare it to the encrypted string in the user account table.

A virtual circuit constitutes a limited functionality session. In the case of identification and authentication, the virtual circuit provides the capability of transferring authentication information between the client and the TOE, and the capability of sending error messages—such as notification of an expired password or authentication failure—to the client without the requirement of establishing a session.

The requirement for a password (and its strength) is controlled through the assigned profile, which the security administrator configures. The Teradata® RDBMS V2R5.0.2 provides for several configurable controls related to password authentication. The following security administrator configurable controls combine to satisfy the requirements regarding passwords:

- Maximum Logon Attempts (with incorrect password)
- Minimum Characters in a Password
- Password Lockout Time

The TOE does not require that a profile be assigned to every user, but if there is no profile assigned, then default security attributes defined in the SysSecDefaults table apply. Further, the secure installation procedures recommend that a profile with password requirements that meet the site policy be implemented and that the security administrator be granted the privilege to manage the profiles.

If a user id is invalid, a session will not be established. If the user id is valid, but the password is invalid, the I&A function will allow up to the configured number of attempts before locking out the account. Each failed logon attempt is logged to the LogOnOff log. The running count of failed logon attempts is maintained in the LockedCount column in the DBC.DBase table.

The data dictionary contains a unique set of security attributes for each user, which includes RoleName. Only an authorized user that has been granted that right by another authorized user (e.g., SecAdmin) can modify these attributes.

The Database uses the GRANT LOGON and/or REVOKE LOGON statements to give specific users permission to log on or to retract permission to log on to the Database from one or more specific client systems, using the *hostid*. The purpose of this parameter is to allow a mainframe-authenticated user to be granted access to the Teradata® RDBMS V2R5.0.2, which implements a trust relationship between the mainframe and the Teradata® Director Program (TDP). The GRANT LOGON statement may be executed with a null password parameter, thus allowing the mainframe authentication mechanism (e.g., RACF or ACF2) to provide database authentication when the user is attaching from the mainframe hostid. Although the parameter exists for LAN-connected clients, the null password mechanism will fail for LAN users, since there is no trust relationship established between the LAN and the Database.

The identification and authentication security function satisfies the following security requirements:

- FIA_AFL.1 Authentication failure handling
- FIA_ATD.1 User attribute definition
- FIA_SOS.1 Verification of secrets
- FIA_UAU.1 Timing of authentication
- FIA_UID.1 Timing of identification
- FIA_USB.1 User-subject binding

6.1.3 User Data Protection

The parser module is directly responsible for discretionary access control. It is responsible for both generating rows in the system access rights table that give a user the right to access an object and for checking of those access rights on subsequent execution of SQL statements.

Each table within the database has an access control list associated with it that defines which users (or groups of users) have which rights to that data. ACLs list the users who are permitted access, and all others are denied. The access rights are distinct privileges. For example, the access right to INSERT into a table does not include the access right to SELECT from a table. If a user is in more than one group, the most permissive right is applied.

The rights to a database object are initially vested in the owner of the object when the object is created. Each object type (e.g., table, database macro) has a predefined set of rights that are granted. The owner can then use SQL GRANT/REVOKE statements to pass these rights on to other users directly, or the rights can be passed to a ROLE and then the use of that role can be given to a set of users.

A user cannot grant more privileges on an object than they have themselves on that object, and the grantee must also have the GRANT privilege (or other applicable privileges) in order to give them to another user. Therefore, if the objective is for only a security administrator to be able to grant rights, then they need to set up the rights to the object so that only the security administrator has the GRANT privilege.

The right to change a database's structure is handled in the same way as rights to access the data contained in the database, and so adheres to the same rules.

The following enforceable rules combine to achieve the access control of database objects (e.g., database structure, database records):

- Only an authorized user can create another user.
- Every object created in the database is uniquely identified in the DBC.TVM table, which identifies the object with a unique primary index pointing to the row entry in the table, thus constituting the data dictionary.
- The TOE enforces Discretionary Access Control (DAC) on objects based on the identity of the user associated with the session.
- The TOE enforces Discretionary Access Control (DAC) on objects based on the following object attributes: (a) the identity of the owner of the object, (b) the object privileges granted on the object, and (c) any security policies in force for the object.
- Privileges are effective in a user session only if: (a) the privilege was granted directly and has not been revoked, (b) the privilege was granted indirectly (e.g., as an "all users" privilege) and has not been revoked, or (c) the privilege was granted to the user via a role and has not been revoked from the role, and the role is still effective in the current session.

The user data protection security function satisfies the following security requirements:

- FDP_ACC.1 Subset access control
- FDP_ACF.1 Security attribute based access control
- FDP_RIP.1 Subset residual information protection

Note: The Teradata® RDBMS V2R5.0.2 offers only limited, well-defined interfaces that ensure users are authenticated and have appropriate access rights prior to allowing data to be accessed and/or updated. In addition, accesses are audited in accordance with the Audit security function (see subsection 6.1.4).

6.1.4 Security Audit

The parser module is directly responsible for enforcement of access control auditing. (As previously mentioned in Section 5, Teradata® RDBMS V2R5.0.2 uses the term "access control" as opposed to "audit" since auditing is done against accesses of database tables.)

Teradata® RDBMS V2R5.0.2 uses the system table DBC.AccLogRuleTbl to determine whether the access request represents one of the table's auditable events. If the access request is one of the auditable events, the Database writes the resulting event(s) to the DBC.AccLogTbl, which is a table that is generated during installation.

The individual events logged are managed through various functional statements (e.g., BEGIN LOGGING), and their parameters (e.g., user name, database name, operation). Each rule allows the event to be entered in the access log table based on "granted access" or "denied access" (or both, since multiple rules can be defined for the same event type, user identity, or object identity).

A series of event logs are used to indicate the startup or shutdown of the audit trail. The DBC.AccLogTbl is created at installation, and secure installation calls for creation of a security administrator role with responsibility for installing the AccRuleTbl macro that allows access logging to be enabled. Also, the DBC is the only user with access rights to the DBC.AccLogTbl at installation, so only this user can grant access rights to other users (including security administrators) to the access log table. Subsequently, a security administrator creates the rules that meet the security policy requirements for auditable event tracking, including, for example, logging changes to the DBC.AccRuleTbl so that changes to the access logging rules are also logged.

The events of a user can be initiated through a BEGIN LOGGING statement that specifies the user name as its parameter. The DBC.AccLogTbl, where the event(s) associated with that user are written, contains the user name as one of its columns.

The user(s) accessing the audit information must have access rights to the DBC.AccLogTbl. At installation, only the DBC user has these rights, so this user grants access only to the security administrator. No other users will have access to read or delete access log table entries.

Searching and sorting audit data is done with SQL statements, and can be done by any user with access rights to the DBC.AccLogTbl. In addition to directly querying the access log table with SQL, authorized users can use the friendlier interface provided with the Teradata® Manager. As with any SQL query, the results set can be ordered by any of the table's columns, which include such fields as LogDate, LogTime, LogonDate, LogonTime, UserName, DatabaseName, etc.

Following is a discussion of what happens when any table, such as the access log table, fills up. There is a set of dictionary tables that are owned by user DBC. The majority of these tables contain the metadata that defines user created tables. There is no space control on these dictionary tables. They expand as long as there is available disk space. The file system divides the disk space into a series of cylinders. A cylinder is a fixed length of contiguous sectors. The number of sectors depends on the disk type. A cylinder contains a set of disk blocks, which is an

encapsulated set of rows for a table. A cylinder contains blocks for either spool tables (temporary results) or permanent tables. When space is needed for a new row, the file system attempts to allocate space in the same disk block with other rows for the table. If none is available, it then attempts to allocate a new disk block on the same cylinder. If no contiguous space is available, it will compact the cylinder, removing any empty space and possibly migrate blocks off of a cylinder in order to free up space. This space compaction and migration may result in allocation of a free cylinder to the table. If there are no free cylinders and all used cylinders have been compacted, then the file cylinder generates an error log entry and resets the system. This reset will free up any cylinders occupied by spool tables so that following the reset new rows could again be inserted into permanent tables until the disk again fills up. Customers normally monitor their disk space so that the disks do not fill up.

The access log tables are protected like all other tables on the database, and so the secure implementation will enable auditing of accesses made to the access log table itself, thus providing the detection capability.

The security audit security function satisfies the following security requirements:

- FAU_GEN.1 Audit data generation
- FAU_GEN.2 User identity association
- FAU_SAR.1 Audit review
- FAU_SAR.3 Selectable audit review
- FAU_SEL.1 Selective audit
- FAU_STG_EXP.1 Protected audit trail storage

6.1.5 Security Management

Access is controlled through the use of user ids and their associated roles, as explained in the access control security function. The Security Management function covers rules that restrict the ability to create and modify the roles, and the behaviors and values given to the functions of those roles. Apart from the reserved user ids (DBC, SYSADMIN, SYSTEMFE), the security administrator can create new user ids (e.g., SecAdmin) with appropriate access privileges in order to more easily manage security in a structured fashion.

The security administrator has responsibility, in a secure implementation, to restrict ordinary users from being granted the rights outlined in Table 5-3. It is possible and preferable to assign a policy of not allowing passwords to expire to user ids assigned to applications.

Likewise, the security administrator has the responsibility of setting up security attribute and TSF data management, in other words, starting with restrictive rights and only granting those rights that meet the criteria of the security policy. A grantor of rights can grant an object privilege only if the grantor is the owner of the object or if the grantor has been granted that object privilege with the GRANT option.

Revoke statements take effect immediately. Since the rights are maintained on disk and in a memory resident cache controlled by the parser, when there is a change to a right, the memory resident cache for all parser occurrences is flushed and the disk-based row is locked from access

until the change is complete. When the parser needs to query a right, it looks in the cache and if it is not resident, it reads the disk-based row and moves it to the cache.

The security management security function satisfies the following security requirements:

- FMT_MOF.1 Management of security functions behavior
- FMT_MSA.1 Management of security attributes
- FMT_MSA.3 Static attribute initialization
- FMT_MTD.1 Management of TSF data

Note: Users can modify their own authentication data (e.g., password); however, according to NIAP interpretation 0346, "users authorized to change their own authentication data are not a distinct role".

- FMT_REV.1 Revocation

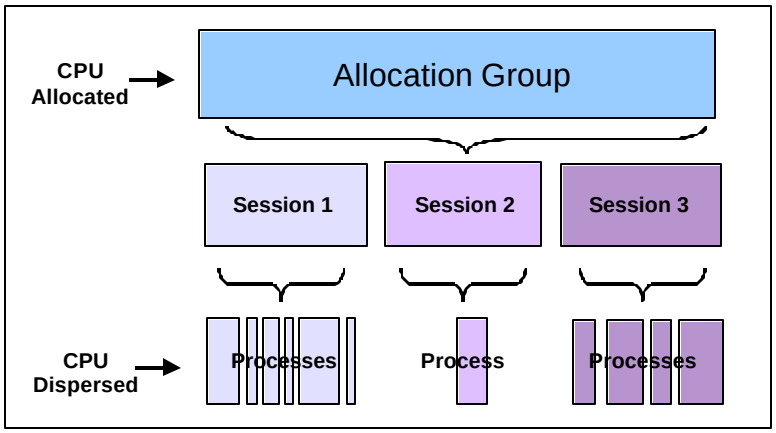
Note: A user can revoke an object privilege from another user, role, or "all users" only if the revoker is the original grantor of the object privilege.

- FMT_SMF.1 Specification of management functions
- FMT_SMR.1 Security roles

6.1.6 Resource Utilization

The Database enforces maximum quotas on various resources to ensure that tables are protected from invalid events that could encroach on valid events.

Figure 6-1 Depiction of CPU Time Allocation by the Teradata â Priority Scheduler



The maximum quota on CPU time is achieved through use of the Teradata® Priority Scheduler, the rules of which are determined by a security administrator who understands the different priorities of the work and has the rights to enact the priority scheduler commands. For example, an interactive user's session is associated with an allocation group that carries a relative weight value that determines the targeted allocation of CPU, as depicted in Figure 6-1. The security

administrator has flexibility in allocating CPU usage between allocation groups, by performance groups, and over performance periods.

The maximum quota on logical data blocks used is enforced using parameters of the "Create User" (or changed using "Modify User") or an associated user profile. The parameters that apply are:

- temp n bytes, which stipulates the size limit that the user can create for the duration of their session in temporary tables, but which is not saved to disk, and
- spool n bytes, which stipulates the size limit that the user can create for forming results sets.

The maximum quota on database storage allocated is also enforced through the following parameter of the "Create User" (or "Modify User") or an associated user profile:

- perm n bytes, which stipulates the size limit for tables that the user can create in a particular database.

The resource utilization function satisfies the following security requirements:

- FRU_RSA.1 Maximum Quotas

6.1.7 Protection of the TSF

The protection of the TSF security function addresses the TSP enforcement functions and in the fact that they must be invoked and succeeded before any other steps can proceed. To begin with, the TDP enforces the rule that, for any one session, there can be only one request outstanding between the TDP and the server at any one time (except for an abort request for a session). In addition, the server never initiates a request.

The only objects on disk that contain "information content" are rows. A row consists of a header and some number of contiguous user-defined columns. When a row is created, columns will either contain a value, input by the user, or have a null value. Columns with a null value do not have any space allocated to them. The only information contained on disk for a null column is a single bit flag in the row header to indicate that the column is null. Columns defined as fixed length are only as wide as the definition. If a user does not input the exact number of columns defined for a fixed length row then the column is padded with space or zero characters. Columns defined as variable length are only as wide as the value input by the user. Therefore, there is no unused space in a new row created in disk. Creating a new row overwrites all of the space allocated for a row. The only time disk resident structures are sent intact outside of the server is for the archive function. In this case a disk block is generated which contains an integral number of rows and no unused space. That block is the unit of transfer to the archive process from the server.

In addition to ensuring that the storage allocated to an object contains no information from any previous use of that storage, the Database also ensures that any privileges on information contained within any object are revoked before the storage for that object is reused for another

object. The parser module generates code to move data into the fixed and variable length columns, to set the presence bits if a column is null, and to set the offsets to the variable length column data. Execution of this code by the database engine task obliterates any data that may have existed in the data space occupied by the row, thus ensuring no object re-use.

The server architecture employs separation of the modules that control security functions (session control, parser, and dispatcher, which make up the PE vproc) from those that do not (the database engine and file manager, which make up the AMP vproc). And since the worker tasks that manipulate data are assigned only after access control has been checked, it is assured that the enforcement function succeeds before the object access is scheduled.

Testing provides confidence that the TSF code and data are protected from external interference and tampering, thus making the TSF self-protecting; for the Database, which is a software only TOE, this is accomplished by the OS, hence the claim of partial self protection. Per the Protection Profile Consistency Guidance, this requirement should be allocated to the IT Environment because the hardware is outside the TOE boundary, and the hardware's OS kernel provides the domain separation.

The Database refers to a Network Time Protocol (NTP) to ensure that all nodes are synchronized across the network. Synchronization with the Teradata® Server is ensured using this NTP. The server itself uses the time that is synchronized between the clients, ensuring that all nodes and the access control entries are synchronized.

The access log tables are protected like all other tables on the Database, and so the secure implementation will enable auditing of accesses made to the access log table itself, thus providing the detection capability.

The protection of the TSF security function satisfies the following security requirements:

- FPT_RVM.1 Non-bypassability of the TSP
- FPT_SEP.EXP.1 TSF domain separation

6.2 ASSURANCE MEASURES

Table 6-1 details the documents that apply to satisfy the Common Criteria EAL2 assurance requirements. All of the documents identified in this table are applicable to the TOE. Some documents may contain further extensions to the referenced TOE version number. For example, V2R5.0.2 indicates Version 2, Major Release 5, Minor Release 0, and Maintenance Level 2. The Maintenance Level identifies modifications were performed on the document for problem isolation. The Rationale column in Table 6-1 provides a more detailed specification and provides a chapter reference in the specific document that contains further details that satisfy the requirement.

Table 6-1 Assurance Requirements Mapped to Documentation

Component	Document(s)	Rationale
ACM_CAP.2: Configuration items	(1) Software Configuration Management (SCM) CMM Practices, March 2001 – 541-0001722-B02 (2) ClearCase Labeling & Branching Standards, March 2003 – 007-0005448-B02 (3) User Guide for ClearCase DBS Development Toolset – 541-0000152-A02 (4) Configuration Item List: • 5.0.2_config.spec.txt • 5.0.2_source.txt	Ref. (1) is the CM plan, which includes the policies and procedures that oversee the NCR CM system. Ref. (2) and (3) provide standards for tying the product's version number to the revision control system. Ref. (4) is the configuration item list that is called for throughout the ACM assurance class, with the first text file being the configuration specification and the second being an itemized list of the source modules.
ADO_DEL.1: Delivery procedures	(1) IPP Quality Check Process, Process No. 1231, Revision No. 001 (2) Procedure for Sub Assembly – 541-0004676-A01 (3) Shipping Procedure – 541-0004677-A01 (4) NCR Teradata® Staging Specification Form, Updated 01 September 2003 (5) pkglist.txt (6) Order Summary, 63059655.xls (7) TWF Maintenance Certification pcitpaW2K702.txt	Ref. (1) is the Solectron South Carolina quality process for IPP Quality Check that ensures that completed Operation Orders are audited and verified before being transferred to Final Assembly (Distribution). Ref. (2) is a Word document that explains the procedure for sub assembly. Ref. (3) is a Word document with flowchart showing the procedure flow for shipping. Ref. (4) defines the requirements, contact information and configurations to be made for staging the deployment of the Teradata® RDBMS V2R5.0.2. Ref. (5) identifies the version numbers for installed packages on the staged system. Ref. (6) is an Excel document that identifies the ProductID and quantity for orders received. Ref. (7) identifies the Windows 2000 Advanced Server platform configuration information for the staged system.

Component	Document(s)	Rationale
ADO_IGS.1: Installation, generation, and start-up procedures	(1) Teradata® RDBMS Release Summary V2R5.0.2 – B035-1098-122A (2) Base System Release Definition V2R5.0.2, October 2003 – B035-1725-093K (3) Upgrading to V2R5.0.2 for W2K, August 2003 – B035-1113-122K (4) WorldMark 4950/5350 Node Software Installation Guide for Microsoft® Windows® 2000 – B035-5540-083K (5) WorldMark® 4475 Software Installation Guide for Microsoft® Windows® 2000 – B035-5913-083K (6) WorldMark® 4455 Software Installation Guide for Microsoft® Windows® 2000 – B035-5902-123E (7) Parallel Upgrade Tool (PUT) for Microsoft® Windows® 2000 User Guide Release 2.0.4 – B035-5710-122K (8) Teradata® RDBMS Security Administration – B035-1100-122E	Ref. (1) provides an overview of the features, enhancement, and RFCs in V2R5.0.2 and includes PDE features. Ref. (2) supports understanding the requirements, dependencies, and support resources for Teradata® RDBMS V2R5.0.2. Ref. (3) explains the steps required to upgrade to V2R5.0.2.x. Ref. (4) explains how to install W2K Advanced Server O/S and Teradata® RDBMS V2R5.0.2 on 4950 or 5350 systems. Ref. (5) explains how to install W2K Advanced Server O/S and Teradata® RDBMS V2R5.0.2 on 4475 systems. Ref. (6) explains how to install W2K Advanced Server O/S and Teradata® RDBMS V2R5.0.2 on 4455 systems. Ref (7) provides instructions for installing/upgrading/using PUT software (the GUI interface for installing/upgrading /configuring Teradata® RDBMS V2R5.0.2 and associated software). Ref. (8) Appendix A of this document provides important information that must be addressed in order to operate the Teradata® RDBMS in a secure state.

Component	Document(s)	Rationale
ADV_FSP.1: Informal functional specification	(1) High Level Design Teradata® Server Architecture Overview, November 2003 – 541-0004657-A03 (2) Teradata® Server Functional Specification, May 11, 2004 – 541-0004655-A03 (3) Introduction to Teradata® RDBMS V2R5.0.2 – B035-1091-122A (4) Teradata® Call-Level Interface Version 2 Reference for Channel-Attached Systems – B035-2417-122A (5) Teradata® Call-Level Interface Version 2 Reference for Network-Attached Systems – B035-2418-122A (6) Teradata® Director Program (TDP) Reference December 2002 – B035-2416-122A (7) Teradata® RDBMS Messages, December 2002 – B035-1096-122A (8) Teradata® RDBMS SQL Reference (Vols. 1,2,4,6) – B035-1101-122A	Ref. (1) fully describes all interfaces to the TSF. Ref. (2) describes the external interfaces to the security functions in an informal manner. Ref. (3) provides an introduction to the Teradata® RDBMS V2R5.0.2. Chapter 2 specifically provides additional details for some of the system components introduced in (2). Chapter 12 specifically provides additional details for the Teradata® RDBMS V2R5.0.2 security controls as introduced in (2). Ref. (4) describes the library of routines that enable a channel-attached application to access data on the Teradata® RDBMS V2R5.0.2, including the error and failure codes returned by the Database. Ref. (5) describes the library of routines that enable an application to access data on the Teradata® RDBMS V2R5.0.2, including the error and failure codes returned by the RDBMS. It provides details for the functions introduced in reference (2). Ref. (6) describes the high-performance communications interface that enables a channel-attached application to communicate with the Teradata® RDBMS V2R5.0.2. Ref. (7) This book lists and explains the messages produced by the Teradata® RDBMS V2R5.0.2 on NCR UNIX MP-RAS systems. It also contains the PDE and Gateway messages for Microsoft Windows 2000 (W2K) systems. Ref. (8) explains relational database concepts to the security administrator and other interested users. Volumes 1 and 4 also provide references for a security administrator to create users, databases, and tables.

Component	Document(s)	Rationale
ADV_HLD.1: Descriptive high-level design	(1) High Level Design Teradata® Server Architecture Overview, November 2003 – 541-0004657-A03 (2) Teradata® Server High Level Design, May 11, 2004 – 541-0004656-A03 (3) Intro to Teradata® RDBMS V2R5.0.2 – B035-1091-122A (4) Teradata® Call-Level Interface Version 2 Reference for Channel-Attached Systems – B035-2417-122A (5) Teradata® Call-Level Interface Version 2 Reference for Network-Attached Systems – B035-2418-122A (6) Teradata® Director Program (TDP) Reference, December 2002 – B035-2416-122A (7) Teradata® RDBMS Messages, December 2002 – B035-1096-122A	Ref. (1) fully describes all subsystems of the TSF. Ref. (2) describes the major components and subcomponents of the Teradata® RDBMS V2R5.0.2 software, and relates those components to the security functions described in the ST. Ref. (3) provides an introduction to the Teradata® RDBMS V2R5.0.2. Chapter 2 specifically provides additional details for some of the system components introduced in (2). Chapter 12 specifically provides additional details for the Teradata® RDBMS V2R5.0.2 security controls as introduced in (2). Ref. (4) describes the library of routines that enable a channel-attached application to access data on the Teradata® RDBMS V2R5.0.2, including the error and failure codes returned by the Database. Ref. (5) describes the library of routines that enable an application to access data on the Teradata® RDBMS V2R5.0.2, including the error and failure codes returned by the RDBMS. Ref. (6) describes the high-performance communications interface that enables a channel-attached application to communicate with the Teradata® RDBMS V2R5.0.2. Ref. (7) This book lists and explains the messages produced by the Teradata® RDBMS V2R5.0.2 on NCR UNIX MP-RAS systems. It also contains the PDE and Gateway messages for Microsoft Windows 2000 (W2K) systems.
ADV_RCR.1: Informal correspondence demonstration	(1) High Level Design Teradata® Server Architecture Overview, November 2003 – 541-0004657-A03 (2) Teradata® Database EAL2 CC Evaluation Representation Correspondence, August 2, 2004 – 541-0004678-A03	Ref. (1) fully describes all subsystems of the TSF. Ref. (2) provides a mapping of the security functions identified in the Security Target to interfaces and enforcement modules defined in the Functional Specification and High-Level Design evidence.

Component	Document(s)	Rationale
AGD_ADM.1: Administrator guidance	(1) Teradata® RDBMS Database Administration V2R5.0.2 – B035-1093-122A (2) Teradata® RDBMS Security Administration V2R5.0.2 – B035-1100-122E (3) Teradata® RDBMS SQL Reference (Vols. 1,2,4,6) – B035-1101-122A (4) Teradata® RDBMS Database Window V2R5.0.2 – B035-1095-122A (5) Introduction to Teradata® RDBMS – B035-1091-122A (6) Teradata® RDBMS Database Design – B035-1094-122A (7) Teradata® RDBMS Utilities (Vols. 1, 2, 3) – B035-1102-122A (8) Teradata® RDBMS Data Dictionary V2R5.0.2 – B035-1092-122A (9) Teradata® RDBMS Performance Optimization V2R5.0.2 – B035-1097-122A (10) Teradata® Archive/Recovery Utility Reference Release 07.00.00 – B035-2412-122A (11) Teradata® Manager User Guide Release 6.0 – B035-2428-122A (12) Teradata® Index Wizard User Guide Release 1.00.00 – B035-2506-122A (13) A Guide to Securing Microsoft Windows 2000, Version 1.1, August 26, 2004	Ref. (1) provides security administrator guidance including the creation, administration, and security of the relational objects. Ref. (2) provides a reference guide for the security administrator in formulating, implementing and auditing a security policy, and explains creation of users, databases, and tables. Ref. (3) explains relational database concepts to the security administrator and other interested users. Volumes 1 and 4 also provide references for a security administrator to create users, databases, and tables. Ref. (4) introduces and explains the Teradata® RDBMS V2R5.0.2 DBW and its commands, which help to operate and maintain the RDBMS. Ref. (5) introduces system administration and security Ref. (6) describes database design for a security administrator. Ref. (7) consists of three volumes of Teradata® RDBMS V2R5.0.2 utility program descriptions. They are used primarily by field engineers, developers, and security administrators. Ref. (8) provides information about the Data Dictionary, including all the tables of the system DBC. Ref. (9) helps a security administrator tune the RDBMS system performance. Ref. (10) is a utility reference for archiving, restoring and recovering databases and tables. Ref. (11) provides “getting started” information for users. Ref. (12) provides information on the Teradata® Index Wizard and includes an overview of the product and its components. It also describes the operational functions and features of the product. Ref. (13) describes the steps that must be performed to harden the underlying Operating System upon which the TOE operates.
AGD_USR.1: User guidance	(1) User Guidance Recap, October 2003 – 541-0004679-A01	Ref. (1) was produced to take the place of user guidance. Teradata® does not produce actual end user guidance documentation. This document maps requirements to documents/chapters that provide the user guidance.

Component	Document(s)	Rationale
ATE_COV.1: Evidence of coverage	(1) Teradata® Server EAL2 CC Evaluation System Test Overview, July 7, 2004 – 541-0004842-A03	Ref. (1) provides an overview of the tests performed by the developer against the functional claims contained in this Security Target.
ATE_FUN.1: Functional testing	(1) Teradata® Server EAL2 CC Evaluation Test Execution Results, July 13, 2004 – 541-0004879-A01 (2) Teradata® Server EAL2 CC Evaluation System Test Overview, July 7, 2004 – 541-0004842-A03 (3) Teradata® Server EAL2 CC Evaluation Test Suite 1, July 6, 2004 – 541-0004843-A03 (4) Teradata® Server EAL2 CC Evaluation Test Suite (Suites 2, 3, 4), December 17, 2003 – 541-0009999-A01 (5) Teradata® Server EAL2 CC Evaluation Test Suite 5, July 6, 2004 – 541-0004847-A03	Ref. (1) identifies the expected and actual results of the security functional testing performed by the developer. Ref. (2) provides an overview of the tests performed by the developer against the claims contained in this Security Target. Ref. (3) defines the specific procedures, inputs, and outputs used by the developer while executing each test case. Ref. (4) defines the specific procedures, inputs, and outputs used by the developer while executing each test case. Ref. (5) defines the specific procedures, inputs, and outputs used by the developer while executing each test case.
ATE_IND.2: Independent testing	(1) Teradata® Server EAL2 CC Evaluation Test Execution Results, July 13, 2004 – 541-0004879-A01 (2) Teradata® Server EAL2 CC Evaluation System Test Overview, July 7, 2004 – 541-0004842-A03 (3) Teradata® Server EAL2 CC Evaluation Test Suite 1, July 6, 2004 – 541-0004843-A03 (4) Teradata® Server EAL2 CC Evaluation Test Suite (Suites 2, 3, 4), December 17, 2003 – 541-0009999-A01 (5) Teradata® Server EAL2 CC Evaluation Test Suite 5, July 6, 2004 – 541-0004847-A03	The evaluation team is responsible for reviewing the developer's functional testing. In addition, they are responsible for the documentation of the activities performed during independent testing. As a result, the documents referenced are utilized as inputs to the evaluation team's testing efforts.
AVA_SOF.1: Strength of TOE security function evaluation	(1) Teradata® Strength of Function Analysis, September 21, 2004 – 541-0004942-A02	The TOE performs user authentication via a password mechanism. This is the only probabilistic / permutational mechanism provided by the TOE and its implementation meets the requirements for SOF-basic.
AVA_VLA.1: Developer vulnerability analysis	(1) Teradata® Database EAL2 CC Evaluation Vulnerability Analysis, June 4, 2004 – 541-0004834-A02	Ref. (1) describes the activities performed by the developer to identify potential vulnerabilities against the TOE and how they have been mitigated. This includes the identification of vulnerabilities available in the public domain as well as those against the TOE evaluated configuration.

6.2.1 Rationale for TOE Assurance Requirements

The TOE stresses assurance through vendor actions that are within the bounds of current best commercial practice. The TOE provides, primarily via review of vendor-supplied evidence, independent confirmation that these actions have been competently performed.

The general level of assurance for the TOE is:

a) Consistent with current best commercial practice for IT development and provides a product that is competitive against non-evaluated products with respect to functionality, performance, cost, and time-to-market, and
b) Meets current constraints on widespread acceptance, by expressing its claims against EAL2 from part 3 of the Common Criteria.

7. PP CLAIMS

This chapter provides detailed information in reference to the Protection Profile conformance identification that appears in Chapter 1, Section 1.10 Protection Profile Conformance.

7.1 PROTECTION PROFILE REFERENCE

This Security Target does not claim Protection Profile conformance.

7.2 PROTECTION PROFILE REFINEMENTS

This Security Target does not claim Protection Profile conformance.

7.3 PROTECTION PROFILE ADDITIONS

This Security Target does not claim Protection Profile conformance.

7.4 PROTECTION PROFILE RATIONALE

This Security Target does not claim Protection Profile conformance.

8. RATIONALE

8.1 SECURITY OBJECTIVES RATIONALE

The rationale that the security objectives are necessary and sufficient to address the Threats to Security, Secure Usage Assumptions, and the Organizational Security Policies is defined in Chapter 4, Security Objectives.

8.2 SECURITY REQUIREMENTS RATIONALE

Table 8-1 demonstrates the mapping of Security Functional Requirements to Security Objectives. Rationale for each mapping is included in the table.

Table 8-1 Mapping of Security Objectives for the TOE to SFRs with Rationale

Security Objectives	Security Functional Requirements	Rationale
O.ACCESS (O.ACCESS.OBJECTS, O.ACCESS.CONTROL, O.ACCESS.RESIDUAL)	FDP_ACC.1, FDP_ACF.1, FDP_RIP.1, FIA_ATD.1, FIA_UID.1, FIA_USB.1, FMT_MSA.1, FMT_MSA.3, FMT_REV.1, FPT_RVM.1, FPT_SEP_EXP.1	O.ACCESS is directly provided by FDP_ACC.1, which defines the access control policy and FDP_ACF.1, which specifies the access control rules. FMT_REV.1 enforces revocation of security attributes. FDP_RIP.1 ensures prevention of access to information residing in database table objects when they are re-allocated to another subject. FIA_USB.1, in conjunction with FIA_ATD.1, ensures the security attributes of a user are bound to subjects created to act on his or her behalf. FIA_UID.1 ensures users are identified prior to any TSF-mediated access actions. FPT_RVM.1 ensures that the traditional reference monitor is always invoked prior to access. FMT_MSA.1 and FMT_MSA.3 provide support for the management of security attributes to control access to database objects. FPT_SEP_EXP.1 assures that another subject cannot intentionally or inadvertently access objects one subject is accessing without a TSF access decision being made for the second subject.
O.ADMIN.TOE	FIA_ATD.1, FIA_USB.1, FMT_MOF.1, FMT_MSA.1, FMT_MTD.1, FMT_SMF.1, FMT_SMR.1, FTA_TSE.1	O.ADMIN.TOE is directly provided by FMT_SMR.1, which provides essential administrative functionality restricted to a security administrator (FMT_MOF.1, FMT_MSA.1, and FMT_MTD.1). FMT_SMF.1 ensures that the TOE is capable of performing the management functions listed in FMT_MOF, FMT_MTD, and FMT_MSA families. FIA_USB.1, in conjunction with FIA_ATD.1, provides support by ensuring that the security attributes of users are associated with subjects acting on the user's behalf. Finally, FTA_TSE.1 allows a security administrator to control session establishment based on client system connection.

Security Objectives	Security Functional Requirements	Rationale
O.AUDIT	FAU_GEN.1, FAU_GEN.2, FAU_SAR.1, FAU_SAR.3, FAU_SEL.1, FAU_STG_EXP.1, FIA_ATD.1, FIA_USB.1, FMT_MTD.1, FPT_STM.1	O.AUDIT is directly provided by FAU_GEN.1, which generates audit records for all security relevant events. FAU_GEN.2, in conjunction with FIA_USB.1, supports the enforcement of individual accountability by ensuring the user responsible for each event can be identified. FIA_ATD.1 provides for the storage of user security attributes. FAU_STG_EXP.1 provides permanent storage for the audit trail, while FMT_MTD.1 provides for protection of that audit trail. FAU_SAR.1 and FAU_SAR.3 provide functions to review the contents of the audit trail, while FAU_SEL.1 provides the ability to select which events are to be audited. Reliable time stamps for the TOE are provided by the underlying operating system, as described in the IT Environment requirement FPT_STM.1
O.I&A.TOE	FIA_AFL.1, FIA_ATD.1, FIA_SOS.1, FIA_UAU.1, FIA_UID.1, FIA_USB.1, FMT_MSA.1, FMT_MTD.1, FTA_TSE.1	O.I&A.TOE is directly provided by FIA_UID.1, which provides the means of identifying users of the TOE. FIA_ATD.1 provides a unique set of user attributes for each user while FMT_MSA.1 and FMT_MTD.1 specify controls over the modification of these attributes. FIA_USB.1 provides an association between these user security attributes with subjects acting on behalf of the user. FTA_TSE.1 controls the ability to create a database session by a user. Additional support for O.I&A.TOE is provided by the addition of Identification and Authentication checks performed by the database. FIA_SOS.1 provides for quality metrics to be applied when new passwords are chosen. FIA_UAU.1 ensures users to be successfully authenticated prior to any TSF-mediated actions. FIA_AFL.1 performs certain actions if a specified number of unsuccessful authentication attempts is exceeded.
O.PARTIAL_SELF_PROTECTION	FPT_SEP_EXP.1	FPT_SEP_EXP.1 is explicitly stated to meet O.PARTIAL_SELF_PROECTION since this is a software only TOE and it cannot be expected to enforce domain separation without the underlying hardware. The requirement FPT_SEP_EXP.1 requires Teradata® RDBMS V2R5.0.2 to protect itself from interference and tampering through its own TSFIs.
O.RESOURCE	FIA_ATD.1, FIA_USB.1, FMT_MTD.1, FRU_RSA.1, FTA_TSE.1	O.RESOURCE is provided by: a) FRU_RSA.1, which provides the means of controlling consumption of resources by individual users (supported by FIA_USB.1 in conjunction with FIA_ATD.1); and b) FTA_TSE.1, which provides the means to deny session establishment; and c) FMT_MTD.1 restricts the control of resource assignment to the security administrator.

Table 8-2 demonstrates complete coverage of the Security Objectives by Security Functional Requirements.

Table 8-2 SFRs Mapped to TOE Security Objectives

	FAU						FDP			FIA						FMT								FPT		FRU	FTA
	GEN.1	GEN.2	SAR.1	SAR.3	SEL.1	STG_EXP.1	ACC.1	ACF.1	RIP.1	AFL.1	ATD.1	SOS.1	UAU.1	UID.1	USB.1	MOF.1	MSA.1	MSA.3	MTD.1	REV.1	SMF.1	SMR.1	RVM.1	SEP_EXP.1	RSA.1	TSE.1
O.ACCESS							X	X	X		X			X	X		X	X		X			X	X		
O.ADMIN.TOE											X				X	X	X		X		X	X				X
O.AUDIT	X	X	X	X	X	X					X				X				X							
O.I&A.TOE										X	X	X	X	X	X		X		X							X
O.PARTIAL_SELF_ PROTECTION																								X		
O.RESOURCE											X				X				X						X	X

8.3 EXPLICITLY STATED SECURITY REQUIREMENTS RATIONALE

This Security Target contains three explicitly stated security functional requirement components, namely FAU_STG_EXP.1, FPT_SEP_EXP.1 and FPT_SEP_ENV_EXP.1. The first component was explicitly stated to provide the ability so specify that the TOE protects stored audit records across the interfaces that are within its control. The latter components were explicitly stated since the security functional requirement component FPT_SEP.1 appearing in Part 2 of the Common Criteria does not provide the ability to specify domain separation enforcement functionality for a software only TOE. The explicitly stated security functional requirement components were created to address this issue and provide the capability to specify a delineation between domain separation functionality provided by a software only TOE (FPT_SEP_EXP.1) and the domain separation functionality relied upon by the underlying operating system and hardware of the TOE (FPT_SEP_ENV_EXP.1).

8.4 TOE SUMMARY SPECIFICATION RATIONALE

Table 8-3 describes the association between the TOE Security Functions and the TOE Security Functional Requirements. This table, in conjunction with rationale provided in Section 6.1, demonstrates that the TOE Security Functional Requirements are satisfied.

Table 8-3 TOE Security Function and SFR Mapping

Security Function	Security Functional Components
TOE Access	FTA_TSE.1 TOE session establishment
Identification and Authentication	FIA_AFL.1 Authentication failure handling
	FIA_ATD.1 User Attribute definition
	FIA_SOS.1 Verification of secrets
	FIA_UAU.1 Timing of authentication
	FIA_UID.1 Timing of identification
	FIA_USB.1 User-subject binding
User Data Protection	FDP_ACC.1 Subset access control
	FDP_ACF.1 Security attribute based access control
	FDP_RIP.1 Subset residual information protection
Security Audit	FAU_GEN.1 Audit data generation
	FAU_GEN.2 User identity association
	FAU_SAR.1 Audit review
	FAU_SAR.3 Selectable audit review
	FAU_SEL.1 Selective audit
	FAU_STG_EXP.1 Protected audit trail storage
Security Management	FMT_MOF.1 Management of security functions behaviour
	FMT_MSA.1 Management of security attributes
	FMT_MSA.3 Static attribute initialisation
	FMT_MTD.1 Management of TSF data
	FMT_REV.1 Revocation
	FMT_SMF.1 Specification of management functions
	FMT_SMR.1 Security roles
Resource Utilization	FRU_RSA.1 Maximum quotas
Protection of the TSF	FPT_RVM.1 Non-bypassability of the TSP
	FPT_SEP_EXP.1 TSF domain separation

Table 8-4 maps each of the SFRs from Chapter 5 to the TSFs defined in Chapter 6, and demonstrates complete coverage of the TOE security functions by security functional requirements.

Table 8-4 SFRs Mapped to TOE Security Functions

	FAU						FDP			FIA						FMT							FPT		FRU	FTA
	GEN.1	GEN.2	SAR.1	SAR.3	SEL.1	STG_EXP.1	ACC.1	ACF.1	RIP.1	AFL.1	ATD.1	SOS.1	UAU.1	UID.1	USB.1	MOF.1	MSA.1	MSA.3	MTD.1	REV.1	SMF.1	SMR.1	RVM.1	SEP_EXP.1	RSA.1	TSE.1
TOE Access																										X
Identification and Authentication										X	X	X	X	X	X											
User Data Protection							X	X	X																	
Security Audit	X	X	X	X	X	X																				
Security Management																X	X	X	X	X	X	X				
Resource Utilisation																									X	
Protection of the TSF																							X	X		

Additional rationale for the TOE Summary Specification is defined in Chapter 6, TOE Security Functions.

8.5 STRENGTH OF FUNCTION RATIONALE

A strength of function analysis is required for security functions in which a qualification of their security behavior can be made using the results of a quantitative or statistical analysis of the security behavior of these mechanisms and the effort required to overcome them. The behavior of the following security functions provided by the Teradata® RDBMS may be realized by probabilistic or permutational mechanisms:

- Enforcement of passwords for FIA_UAU.1 Timing of authentication

8.5.1 Enforcement of passwords

The Teradata® RDBMS requires that a password be provided in order to authenticate a user prior to establishing a database session. This mechanism meets or exceeds a rating of SOF-basic in that the mechanism provides adequate protection against casual breach of TOE security by attackers possessing a low attack potential.

The following password controls are implemented:

- minimum number of characters required in a valid password string - `passwordminchar = 8` characters

- number of erroneous sequential logon attempts a user is allowed before the user is locked to further logon attempts – `maxlogonattempts = 3` attempts
- user lock time duration after the user has exceeded the maximum number of logon attempts - `lockeduserexpire = 5` minutes
- time span during which a password is valid - `expirepassword = 90` days

The Teradata® RDBMS can accept any string for a password as long as it conforms to Teradata® RDBMS rules for a word. These rules include that the string may contain:

- Any of 26 alphabetic characters A-Z (case insensitive)
- Any of 10 numeric characters 0 –9
- Any of 3 special characters - $ (dollar sign), _ (underscore), # (pound sign)

The first character of the password string must not be a numeric character.

Assuming that a user constructs a password using the minimum number of characters required, the total number of valid password combinations can be calculated as follows:

26(39)(39)(39)(39)(39)(39)(39) = 3,568,006,173,654

The average number of passwords that an attacker would have to enter before guessing the correct password can be calculated as:

3,568,006,173,654 / 2 = 1,784,003,086,827

The password enforcement mechanism is designed with a user lockout feature. Upon the third failed authentication attempt, the user is locked out for five minutes. As such, an attacker can, at best, attempt three logons every five minutes, or 36 logon attempts per hour.

The number of days required to conduct a successful attack can thus be calculated as follows:

1,784,003,086,827 / 36 / 24 = ~2,064,818,387

This far exceeds the 90-day time span for which a password would be valid. This would indicate that the password enforcement mechanism is at least resistant to an attacker with low attack potential and would meet or exceed the SOF-basic claim.

Assuming a worse case scenario in which a user might construct a password using only one numeric or special character, the total number of valid password combinations would then be calculated as follows:

26(26)(26)(26)(26)(26)(26)(13) = 104,413,532,288

The average number of passwords that an attacker would have to enter before guessing the correct password would be calculated as:

104,413,532,288 / 2 = 52,206,766,144

The number of days required to conduct a successful attack would thus be calculated as follows:

52,206,766,144 / 36 / 24 = ~60,424,497

Again, this far exceeds the 90-day time span for which a password would be valid and would indicate that the password enforcement mechanism is at least resistant to an attacker with low attack potential in order to meet or exceed the SOF-basic claim.

8.6 PP CLAIMS RATIONALE

The rationale for the Protection Profile conformance claims is defined in Chapter 7, Protection Profile Rationale.

APPENDIX A—ACRONYM LIST

AMP	Access Module Processor
AWS	Administrative Workstation
CLI	Call Level Interface
CNS	Console Subsystem
CPU	Central Processing Unit
DBW	Database Window
LAN	Local Area Network
MOSI	Micro Operating System Interface
MTDP	Micro Teradata® Director Program
OS	Operating System
PDE	Parallel Database Extension
PE	Parsing Engine
RAID	Redundant Array of Independent Disks
RDBMS	Relational Database Management System
SQL	Structured Query Language
TDP	Teradata® Director Program
TPA	Trusted Parallel Application
vdisk	Virtual Disk
vproc	Virtual Processor