EMC Corporation
EMC® Greenplum® 4.2

Security Target

Evaluation Assurance Level (EAL): EAL2+
Document Version: 0.9

Prepared for:

EMC Corporation
176 South Street
Hopkinton, MA 01748
United States of America

Phone: +1 508 435 1000
http://www.emc.com

Prepared by:

Corsec Security, Inc.
13135 Lee Jackson Memorial Highway, Suite 220
Fairfax, VA 22033
United States of America

Phone: +1 703 267 6050
http://www.corsec.com

Table of Contents

1 INTRODUCTION 4
1.1 PURPOSE 4
1.2 SECURITY TARGET AND TOE REFERENCES 4
1.3 PRODUCT OVERVIEW 5
1.4 TOE OVERVIEW 7
1.4.1 TOE Environment 8
1.5 TOE DESCRIPTION 9
1.5.1 Physical Scope 9
1.5.2 Logical Scope 11
1.5.3 Product Physical and Logical Features and Functionality not included in the TOE 12

2 CONFORMANCE CLAIMS 13

3 SECURITY PROBLEM 14
3.1 THREATS TO SECURITY 14
3.2 ORGANIZATIONAL SECURITY POLICIES 15
3.3 ASSUMPTIONS 15

4 SECURITY OBJECTIVES 17
4.1 SECURITY OBJECTIVES FOR THE TOE 17
4.2 SECURITY OBJECTIVES FOR THE OPERATIONAL ENVIRONMENT 18
4.2.1 IT Security Objectives 18
4.2.2 Non-IT Security Objectives 19

5 EXTENDED COMPONENTS 20
5.1 EXTENDED TOE SECURITY FUNCTIONAL COMPONENTS 20
5.1.1 Class FAU: Security Audit 20
5.1.2 Class FMT: Security Management 21
5.1.3 Class FPT: Protection of the TSF 22
5.1.4 Class FTA: TOE Access 22
5.2 EXTENDED TOE SECURITY ASSURANCE COMPONENTS 23

6 SECURITY REQUIREMENTS 24
6.1 CONVENTIONS 24
6.2 SECURITY FUNCTIONAL REQUIREMENTS 24
6.2.1 Class FAU: Security Audit 26
6.2.2 Class FDP: User Data Protection 29
6.2.3 Class FMT: Security Management 31
6.2.4 Class FPT: Protection of the TSF 33
6.2.5 Class FRU: Resource Utilization 34
6.2.6 Class FTA: TOE Access 35
6.3 SECURITY ASSURANCE REQUIREMENTS 36

7 TOE SUMMARY SPECIFICATION 37
7.1 TOE SECURITY FUNCTIONS 37
7.1.1 Security Audit 38
7.1.2 User Data Protection 40
7.1.3 Identification and Authentication 41
7.1.4 Security Management 41
7.1.5 Protection of the TSF 42
7.1.6 Resource Utilization 42
7.1.7 TOE Access 42

8 RATIONALE 44
8.1 CONFORMANCE CLAIMS RATIONALE 44
8.1.1 Protection Profile Conformance 44

8.2 SECURITY OBJECTIVES RATIONALE 44
8.2.1 Security Objectives Rationale Relating to Threats 44
8.2.2 Security Objectives Rationale Relating to Policies 50
8.2.3 Security Objectives Rationale Relating to Assumptions 50
8.3 RATIONALE FOR EXTENDED SECURITY FUNCTIONAL REQUIREMENTS 53
8.4 RATIONALE FOR EXTENDED TOE SECURITY ASSURANCE REQUIREMENTS 54
8.5 SECURITY REQUIREMENTS RATIONALE 54
8.5.1 Rationale for Security Functional Requirements of the TOE Objectives 54
8.5.2 Security Assurance Requirements Rationale 59
8.5.3 Rationale for Security Assurance Requirements of the TOE Objectives 60
8.5.4 Dependency Rationale 63

9 ACRONYMS AND TERMS 66
9.1 ACRONYMS 66
9.2 TERMS 67

Table of Figures

FIGURE 1 – DEPLOYMENT CONFIGURATION OF THE TOE 8
FIGURE 2 – PHYSICAL TOE BOUNDARY 10
FIGURE 3 – SECURITY AUDIT DATA GENERATION FAMILY DECOMPOSITION 20
FIGURE 4 – SECURITY MANAGEMENT FAMILY DECOMPOSITION 21
FIGURE 5 – PROTECTION OF THE TSF FAMILY DECOMPOSITION 22
FIGURE 6 – TOE ACCESS HISTORY FAMILY DECOMPOSITION 23

List of Tables

TABLE 1 – ST AND TOE REFERENCES 5
TABLE 2 – TOE ENVIRONMENT MINIMUM REQUIREMENTS 9
TABLE 3 – CC AND PP CONFORMANCE 13
TABLE 4 – THREATS 14
TABLE 5 - ORGANIZATIONAL SECURITY POLICIES 15
TABLE 6 – ASSUMPTIONS 16
TABLE 7 – SECURITY OBJECTIVES FOR THE TOE 17
TABLE 8 – IT SECURITY OBJECTIVES 18
TABLE 9 – NON-IT SECURITY OBJECTIVES 19
TABLE 10 – TOE SECURITY FUNCTIONAL REQUIREMENTS 24
TABLE 11 – AUDITABLE EVENTS 26
TABLE 12 – ASSURANCE REQUIREMENTS 36
TABLE 13 – MAPPING OF TOE SECURITY FUNCTIONS TO SECURITY FUNCTIONAL REQUIREMENTS 37
TABLE 14 – AUDIT RECORD CONTENTS 39
TABLE 15 – ROLE ATTRIBUTES 40
TABLE 16 – OBJECT PRIVILEGES 41
TABLE 17 – THREATS:OBJECTIVES MAPPING 44
TABLE 18 - POLICIES:OBJECTIVES MAPPING 50
TABLE 19 – ASSUMPTIONS:OBJECTIVES MAPPING 51
TABLE 20 – RATIONALE FOR EXTENDED REQUIREMENTS 53
TABLE 21 – OBJECTIVES:SFRS MAPPING 55
TABLE 22 – OBJECTIVES:SARS MAPPING 60
TABLE 23 – FUNCTIONAL REQUIREMENTS DEPENDENCIES 63
TABLE 24 – ACRONYMS 66

1 Introduction

This section identifies the Security Target (ST), Target of Evaluation (TOE), and the ST organization. The Target of Evaluation (TOE) is the EMC® Greenplum® 4.2, and will hereafter be referred to as the TOE throughout this document. The TOE is a data analysis platform and Relational Database Management System (RDBMS). Data is distributed across and held in multiple physically separate nodes, resulting in a distributed architecture. Each node additionally offers multi-core computing capabilities, allowing the TOE to distribute data analysis jobs across multiple nodes for parallel processing. The end result is a flexible, scalable, and powerful data analysis platform.

This ST conforms to the following protection profile:

- U.S. Government Protection Profile for Database Management Systems, Version 1.3, December 24, 2010

The above protection profile will be referred to throughout this ST using the following shortened name and acronym:

- Short name: Database Management Systems PP, Acronym: DBMS

This protection profile requires assurance at EAL 2, augmented by ALC_FLR.2. This Security Target claims all of the functional requirements needed to conform to the DBMS PP.

1.1 Purpose

This ST is divided into nine sections, as follows:

- Introduction (Section 1) – Provides a brief summary of the ST contents and describes the organization of other sections within this document. It also provides an overview of the TOE security functions and describes the physical and logical scope for the TOE, as well as the ST and TOE references.
- Conformance Claims (Section 2) – Provides the identification of any Common Criteria (CC), Protection Profile, and EAL package claims. It also identifies whether the ST contains extended security requirements.
- Security Problem (Section 3) – Describes the threats, organizational security policies, and assumptions that pertain to the TOE and its environment.
- Security Objectives (Section 4) – Identifies the security objectives that are satisfied by the TOE and its environment.
- Extended Components (Section 5) – Identifies new components (extended Security Functional Requirements (SFRs) and extended Security Assurance Requirements (SARs)) that are not included in CC Part 2 or CC Part 3.
- Security Requirements (Section 6) – Presents the SFRs and SARs met by the TOE.
- TOE Summary Specification (Section 7) – Describes the security functions provided by the TOE that satisfy the security functional requirements and objectives.
- Rationale (Section 8) - Presents the rationale for the security objectives, requirements, and SFR dependencies as to their consistency, completeness, and suitability.
- Acronyms (Section 9) – Defines the acronyms and terminology used within this ST.

1.2 Security Target and TOE References

Table 1 below shows the ST and TOE references.

Table 1 – ST and TOE References

ST Title	EMC Corporation EMC® Greenplum® 4.2 Security Target
ST Version	Version 0.9
ST Author	Corsec Security, Inc.
ST Publication Date	March 22, 2012
PP Identification	U.S. Government Protection Profile for Database Management Systems, Version 1.3, December 24, 2010
TOE Reference	EMC® Greenplum® 4.2.0.0 build 5

1.3 Product Overview

The Product Overview provides a high level description of the product that is the subject of the evaluation. The following section, TOE Overview, will provide the introduction to the parts of the overall product offering that are specifically being evaluated.

The EMC® Greenplum® 4.2 database offers two main features:

1. An RDBMS (based on a modified PostgreSQL database) that allows customers to manage the data in the terabyte-to-petabyte range within the database using standard Structured Query Language (SQL) commands.
2. A distributed processing framework called MapReduce. MapReduce allows customers to run complex data analysis on data contained within the database using their preferred programming languages (of the ones supported). This frees customers from the limitations of having to use ordinary SQL, and allows them to leverage the flexibility of languages like C++ and Perl while working directly with data stored in the database.

These features are implemented on a distributed architecture that allows the EMC® Greenplum® 4.2 platform to manage workloads across many computers with multiple processing cores. Users submit analysis jobs to a component called the master host (or master), which is a physically separate computer with its own hardware, software, and Operating System (OS)[1].

In addition to the master, the product contains a series of segment servers (also with their own hardware, software, and OS) that hold data and perform processing for the master upon request. The master communicates with segment servers via a private network that is not accessible to users. The network is composed of a set of Internet Protocol (IP) switches that interconnect all of the segment servers and the master. Data is broken apart at the table level to maximize the efficiency of the distributed model. This means that a single table may have rows spread across many segments, allowing each segment to apply its processing capabilities simultaneously to perform operations on the data.

Users and administrators access the product via a client program that can be either a command line or graphical tool. The client program can be pre-built or custom-built by the customer based on the Open Database Connectivity (ODBC) and Java Database Connectivity (JDBC) standards. All system permissions and configuration parameters are stored and modified in the database, along with user data. This means that both users and administrators connect via the same interface. This interface is used to manage and configure the system, manage user data, and to submit jobs to be run on the system[2].

[1] See Table 2 for a list of supported platforms.

[2] Administrators can configure individual host settings (such as host name, IP address, etc.) and troubleshoot servers via direct access to a Linux Bourne Again Shell (BASH), accessed via Secure Shell (SSH).

All end-users and administrators are assigned roles with customizable sets of permissions; only those roles assigned to administrators have administrative permissions. The system also defines a SuperUser role that has access to everything on the system. Users authenticate with the master before they are assigned a role, then the master acts on behalf of users (while assuming the user's role) to coordinate jobs.

The system also provides a web-based Graphical User Interface (GUI) called Greenplum Performance Monitor Console. The performance monitor provides a web interface that administrators can use to view system performance. This GUI is housed on the master, and administrators can connect to it via a standards-compliant web browser with graphical support. Although the Performance Monitor Console is used exclusively to view performance data, performance data is stored in the database and can also be accessed via a client application connected to the master.

The master uses an active queuing system to determine when to run user jobs. Every user account—except for superuser administrator accounts—is assigned a queue where their jobs are placed. Every queue has a configurable limit of the maximum number of jobs and number of resources that can be used. Once these limits are reached, the master puts any additional jobs from a user on the same queue on hold. Jobs on hold remain idle until active jobs complete or the number of resources in use drops under the defined thresholds. This queuing system allows the system to prioritize jobs within each queue over others (since priorities are assigned to each queue), while preventing the system from being overwhelmed by a flood of jobs.

The master is responsible for controlling and distributing jobs to the segment servers, and consolidating the results to return to the user. Once a job is ready to run, the master distributes the job to the segment servers. The master maintains a set of system tables that describe the way data is structured and distributed within the database called a system catalog. The master uses a system catalog to determine how to distribute jobs. Users can submit jobs using SQL statements or one of the three languages supported by MapReduce: Perl, Python, and C. The results of jobs are returned to the master in pieces as each segment server completes its own share of the processing. The master reassembles the pieces and interprets the results to present to the user's client program.

The system is capable of loading data that has been extracted from external sources (such as data from external databases or flat files) if the data is in either the delimited text or comma separated values format. The system can also work with these external data sources as "external tables". External tables can be read-only for data loading (the process of getting data onto the Greenplum database) or writable for data unloading (the process of getting data out of the Greenplum database). The external tables functionality is also used to mount log files so that the data can be operated on via SQL statements.

Data loading and unloading are much faster than with traditional RDBMS products due to the distributed architecture. The Greenplum database supports data loading and unloading either infrequently for large amounts of data or frequently for small updates.

To help cope with large volumes of data, the Greenplum database supports data compression via one of the supported compression methods (zlib, QuickLZ, or TBD). The system also supports expansion of the database onto additional segment servers. This is accomplished by reapportioning user data across the new segments. Since individual tables are spread across multiple segments, this reapportionment is necessary to maximize the capabilities of the distributed architecture.

Each component of the product (the master, the segments, and the private network) is redundant to support high availability. Every segment is additionally capable of being mirrored across segment servers to prevent a single point of failure from causing data loss. The master synchronizes regularly with a backup master, and an administrator can manually switch the backup master to take over in the event of a major failure.

1.4 TOE Overview

The TOE Overview summarizes the usage and major security features of the TOE. The TOE Overview provides a context for the TOE evaluation by identifying the TOE type, describing the product, and defining the specific evaluated configuration.

The software-only TOE is the EMC® Greenplum® 4.2 data analysis platform. The TOE includes a RDBMS combined with MapReduce functionality and a standards-compliant (ODBC and JDBC) interface to run queries and data analysis jobs. The software is designed to be distributed across multiple physical nodes, but maintains the functionality of a single RDBMS. Users access the RDBMS via the master, while the TOE stores data on segments. The master also stores a data catalog that records where data is distributed across the segments.

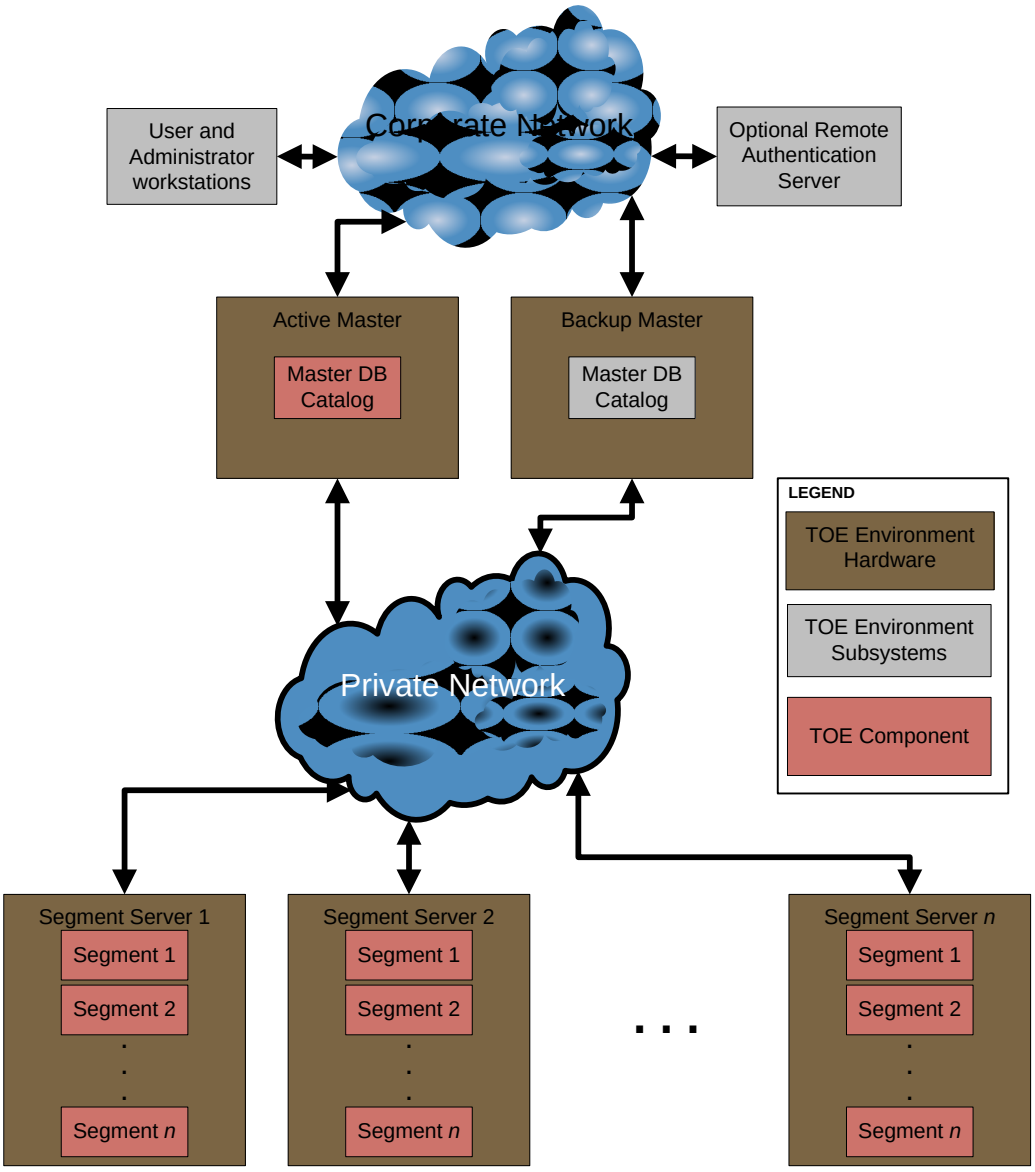
Although the hardware is not a part of the TOE, the recommended number of segments is determined by the number of Central Processing Unit (CPU) cores on each segment server. The number of cores must be divisible by the number of segments, so for a twelve-core system you could have 4, 6, or 12 active segments per host. Each platform can undergo performance testing to determine the optimal number of segments. In contrast, there is only ever one active master at any given time, regardless of the number of segments. Each master and segment runs at least one instance of the Greenplum database process.

Users authenticate with the master either via the TOE's local authentication process or via a remote authentication server. After successfully authenticating, users and administrators can send jobs to the master to be run by the TOE. The master deconstructs and optimizes each job and rewrites the job into a query plan. The query plan is a distributable version of the job that the master builds, and includes all segments that are to run the job. The master then sends the entire query plan to each segment. Segments perform the relevant processing and return any results to the master.

Segments run a file replication process that creates a mirror copy of the data on the segment and moves the data to another segment server. Mirroring of segment data prevents the TOE from suffering data loss in the event that a segment fails and cannot be recovered.

Figure 1 shows the details of the deployment configuration of the TOE. The red elements in the figure are TOE components. The following previously-undefined acronym appears in Figure 1:

- DB – Database



Segment Servers contain one primary segment per one or more CPU cores

Figure 1 – Deployment Configuration of the TOE

1.4.1 TOE Environment

The TOE is a distributed RDBMS designed to run on general-purpose commodity hardware. The OS that the TOE runs on is included within the installation image and installed as part of the process of installing the TOE.

In addition to hardware, the TOE needs the following environmental components in order to function properly:

- cables, connectors, and switching and routing devices that allow all of the TOE and environmental components to communicate with each other
- an administrator workstation with a standards-compliant client program

The TOE is intended to be deployed in a secure data center that protects physical access to the TOE. The TOE is intended to be interconnected by a back-end private network that does not connect directly to external hosts.

The TOE provides access control to an RDBMS with analytic capabilities. Some of the available access control mechanisms (such as Lightweight Directory Access Protocol (LDAP)) require the use of a remote authentication server. The TOE environment is required to provide this.

The TOE is capable of loading data from external sources and unloading data to external sources (such as external files or named pipes to be imported into a data warehouse). It is the responsibility of the administrators and the TOE environments to ensure that such data is accurate and not modified in transit. Additionally, administrators should ensure that any external data is in a format compatible with the TOE before attempting to load it onto the TOE.

Table 2 below specifies the minimum system requirements for the proper operation of the TOE.

Table 2 – TOE Environment Minimum Requirements

Operating System	Red Hat Enterprise Linux (RHEL) v5.5, v5.7, and v6.1
File Systems	xfs required for data storage Red Hat (ext3 supported for root file system)
Minimum CPU	Pentium Pro compatible (P3/Athlon and above)
Minimum Memory	16 GB[3] RAM[4] per server
Disk Requirements	• 150 MB[5] per host for Greenplum installation • Approximately 300 MB per segment instance for meta data • Appropriate free space for data with disks at no more than 70% capacity • High-speed local storage
Network Requirements	Gigabit Ethernet within the array Dedicated, non-blocking switch
Software and Utilities	bash shell GNU's Not Unix (GNU) tar GNU zip GNU Compiler Collection (GCC) runtime libraries (glibc, etc., but not including the GCC compiler) GNU readline (Solaris only)[6]

1.5 TOE Description

This section primarily addresses the physical and logical components of the TOE included in the evaluation.

1.5.1 Physical Scope

Figure 2 illustrates the physical scope and the physical boundary of the overall solution and ties together all of the components of the TOE.

[3] GB – Gigabyte
[4] RAM – Random Access Memory
[5] MB – Megabyte
[6] On Solaris platforms, GNU Readline is required to support interactive administrative utilities.

The TOE is an RDBMS and data analysis platform that runs on general-purpose hardware compliant to the minimum software and hardware requirements as listed in Figure 2. The TOE is installed on distributed hardware, as depicted in the figure below. The essential physical components for the proper operation of the TOE in the evaluated configuration are:

- the Greenplum software.

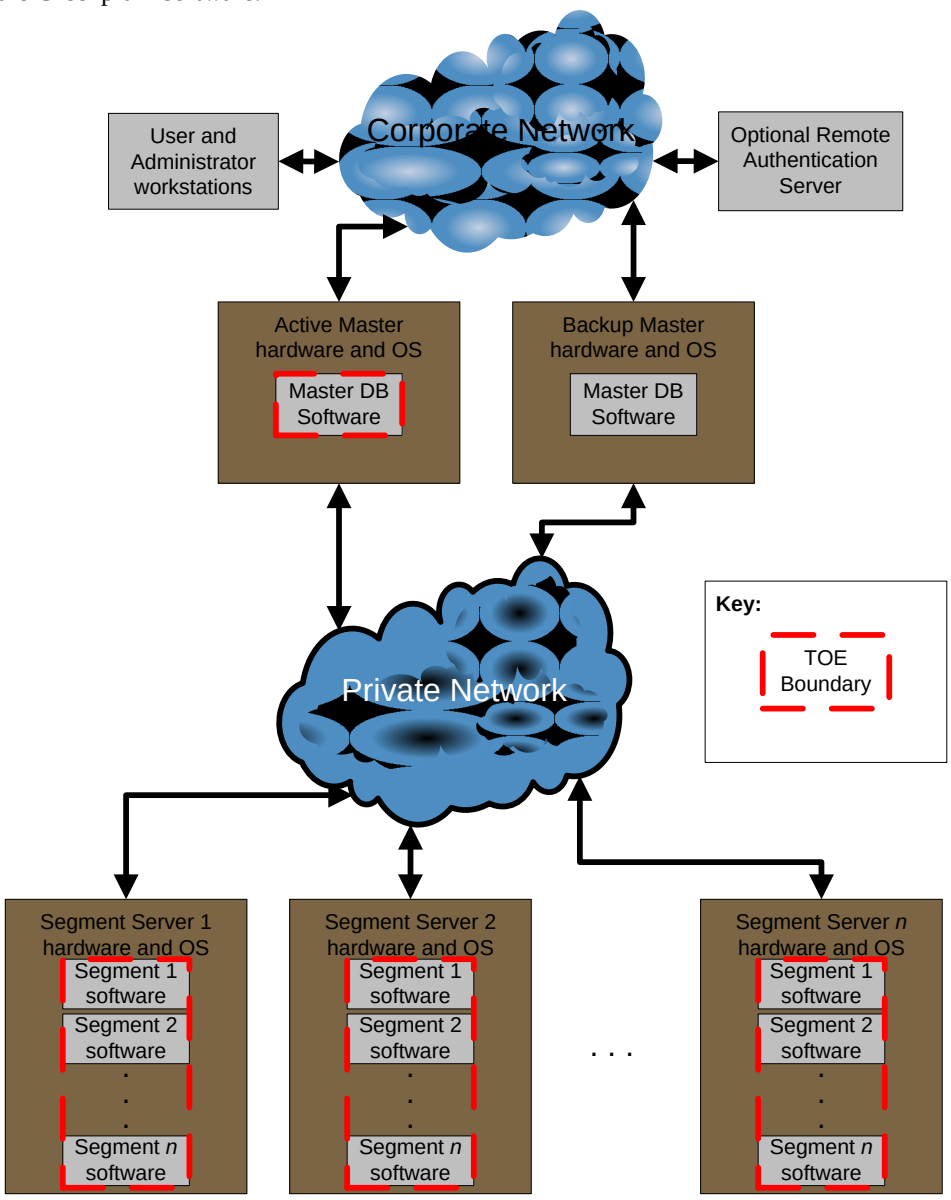


Figure 2 – Physical TOE Boundary

In Figure 2 above, the different boxes with Segment software and Master DB Software represent different instantiations of the same software.

1.5.1.1 TOE Software

The TOE is a software-only TOE meant to be used with commodity hardware.

1.5.1.2 Guidance Documentation

The following guides are required reading and part of the TOE:

- Greenplum® Database 4.2 Installation Guide
- Greenplum® Database 4.2 Administration Guide
- Greenplum® Database 4.2 Release Notes

1.5.2 Logical Scope

The logical boundary of the TOE will be broken down into the following security classes which are further described in sections 6 and 7 of this ST. The logical scope also provides the description of the security features of the TOE. The security functional requirements implemented by the TOE are usefully grouped under the following Security Function Classes:

- Security Audit,
- User Data Protection,
- Identification and Authentication,
- Security Management,
- Protection of the TOE Security Functionality (TSF),
- Resource Utilization, and
- TOE Access.

1.5.2.1 Security Audit

The TOE generates audit records for startup and shutdown of the system, segment database failures, SQL statements that result in an error, informational information on SQL statements, and all connection attempts and disconnects (if configured). Administrators can select events to be excluded from audit. Since audit data is stored in the database, administrators can run a variety of SQL commands to facilitate viewing, searching, sorting, and ordering of audit data. Audit data is also stored in log files, but these can be accessed via SQL queries. Audit data is protected from unauthorized deletion and modification by role-based permissions and access control.

1.5.2.2 User Data Protection

The TOE controls access to user data via a Data Access Security Functional Policy (SFP). The Data Access SFP relies on role-based permissions and built-in access control mechanisms to ensure that only authorized users can access data. Additionally, access to data can be controlled by defining views that only show a portion of the data contained within a table and granting users access to the views instead of tables. Access to data can be granted or revoked by administrators or other authorized users. There is no shared memory between user's processes.

1.5.2.3 Identification and Authentication

Identification and authentication of users and administrators is performed by the master. Users and administrators submit login credentials via a client application. Once successfully authenticated, the master binds to the user or administrator's account and assumes the account's role (permissions) when submitting jobs on behalf of the user or administrator. No tasks can be performed by users or administrators until they have successfully authenticated with the TOE.

The TOE stores credentials within the database, but remote authentication methods are supported. The TOE also records the authentication type and any roles associated with the account.

1.5.2.4 Security Management

Management of the TOE is provided via the same interface that is used to access user data. All configuration parameters are stored in the database, and can be used to determine the behavior of the TOE

security functions. Access via client applications can be command line or graphical, depending on the client application used. Management access is limited by role.

1.5.2.5 Protection of the TSF

The TOE master component constantly synchronizes with a backup. In the event that the master fails, no additional tasks can be executed, and already-running jobs fail and roll back to their previous state. The master can be restored when an administrator manually sets the backup master to take over. Synchronization data is sent over the private network and is inaccessible to TOE users.

1.5.2.6 Resource Utilization

As mentioned above, the segments fail and roll back to a previous state upon failure of a master node. On a fully functional system, the master is able to limit the number of simultaneous jobs or the total processing cost available to users. This is done by assigning a queue to each user or group of users and enforcing maximum quotas on each queue. Once a queue reaches its assigned threshold, all additional jobs are set to idle until currently-running jobs finish.

1.5.2.7 TOE Access

The TOE defines an administrator-configurable parameter for each user that can limit the number of concurrent sessions for the account. Once this limit is reached, the TOE denies the establishment of any additional session by the user. The TOE also denies session establishment of any user account where the password field is given a null value. Upon request, the TOE can display the date and time of the last successful and unsuccessful login to the administrator.

1.5.3 Product Physical and Logical Features and Functionality not included in the TOE

Features and functionality that are not part of the evaluated configuration of the TOE are:

- All hardware and operating systems upon which the TOE runs (including BASH).
- Greenplum Performance Monitor

2 Conformance Claims

This section and Table 3 provide the identification for any CC, Protection Profile (PP), and EAL package conformance claims. Rationale is provided for any extensions or augmentations to the conformance claims. Rationale for CC and PP conformance claims can be found in Section 8.1.

Table 3 – CC and PP Conformance

Common Criteria (CC) Identification and Conformance	Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 3, July 2009; CC Part 2 extended; CC Part 3 conformant; PP claim Database Management Version 1.3; Parts 2 and 3 Interpretations of the Common Evaluation Methodology (CEM) as of 2011-02-09 were reviewed, and no interpretations apply to the claims made in this ST.
PP Identification	U.S. Government Database Management Protection Profile, Version 1.3, December 24, 2010.
Evaluation Assurance Level	EAL2+ augmented with Flaw Remediation (ALC_FLR.2)

3 Security Problem

This section describes the security aspects of the environment in which the TOE will be used and the manner in which the TOE is expected to be employed. It provides the statement of the TOE security environment, which identifies and explains all:

- Known and presumed threats countered by either the TOE or by the security environment
- Organizational security policies with which the TOE must comply
- Assumptions about the secure usage of the TOE, including physical, personnel and connectivity aspects

3.1 Threats to Security

This section identifies the threats to the IT[7] assets against which protection is required by the TOE or by the security environment. The threat agents are divided into two categories:

- Attackers who are not TOE users: They have public knowledge of how the TOE operates and are assumed to possess a low skill level, limited resources to alter TOE configuration settings or parameters and no physical access to the TOE.
- TOE users: They have extensive knowledge of how the TOE operates and are assumed to possess a high skill level, moderate resources to alter TOE configuration settings or parameters and physical access to the TOE. (TOE users are, however, assumed not to be willfully hostile to the TOE.)

Both are assumed to have a low level of motivation. The IT assets requiring protection are the TSF[8] and user data saved on the TOE. Removal, diminution and mitigation of the threats are through the objectives identified in Section 4 Security Objectives. Table 4 below lists the applicable threats.

Table 4 – Threats

Name	Description
T.ACCOUNTABILITY	An administrator might not be able to determine the user responsible for malicious actions that degrade TOE functions.
T.AUDIT_COMPROMISE	A user or process may view audit records, cause audit records to be lost or modified, or prevent future audit records from being recorded, thus masking a user's action.
T.MASQUERADE	A user or process may masquerade as another entity in order to gain unauthorized access to data or TOE resources.
T.UNAVAILABILITY	The TOE may be overwhelmed by legitimate user tasks, preventing or delaying any TOE functionality from being accessed.
T.TSF_COMPROMISE	A malicious user or process may cause configuration data to be inappropriately accessed (viewed, modified or deleted).
T.UNAUTHORIZED_ACCESS	A user may gain unauthorized access to user data for which they are not authorized according to the TOE security policy.
T.UNIDENTIFIED_ACTIONS	Failure of the authorized security administrator to identify and act upon unauthorized actions may occur.

[7] IT – Information Technology
[8] TSF – TOE Security Functionality

Name	Description
T.CRITICAL_FAILURE	The TOE may experience a failure of a critical component that prevents users and administrators from being able to access TOE functionality.
T.ACCIDENTAL_ADMIN_ERROR	An administrator may incorrectly install or configure the TOE resulting in ineffective security mechanisms.
T.POOR_DESIGN	Unintentional errors in requirements specification or design of the TOE may occur, leading to flaws that may be exploited by a casually mischievous user or program.
T.POOR_IMPLEMENTATION	Unintentional errors in implementation of the TOE design may occur, leading to flaws that may be exploited by a casually mischievous user or program.
T.POOR_TEST	Lack of or insufficient tests to demonstrate that all TOE security functions operate correctly (including in a fielded TOE) may result in incorrect TOE behavior being discovered thereby causing potential security vulnerabilities.
T.RESIDUAL_DATA	A user or process may gain unauthorized access to data through reallocation of TOE resources from one user or process to another.

3.2 Organizational Security Policies

An Organizational Security Policy (OSP) is a set of security rules, procedures, or guidelines imposed by an organization on the operational environment of the TOE. Table 5 below lists the OSPs that are presumed to be imposed upon the TOE or its operational environment by any organization implementing the TOE in the CC evaluated configuration.

Table 5 - Organizational Security Policies

Name	Description
P.ACCOUNTABILITY	The authorized users of the TOE shall be held accountable for their actions within the TOE.
P.ROLES	The TOE shall provide an authorized administrator role for secure administration of the TOE. This role shall be separate and distinct from other authorized users.

3.3 Assumptions

This section describes the security aspects of the intended environment for the evaluated TOE. The operational environment must be managed in accordance with assurance requirement documentation for delivery, operation, and user guidance. Table 6 lists the specific conditions that are required to ensure the security of the TOE and are assumed to exist in an environment where this TOE is employed.

Table 6 – Assumptions

Name	Description
A.DOMAIN_SEPARATION	The operational environment will provide a separate domain for the TOE's operation.
A.I&A	The operational environment will provide identification and authentication mechanisms for use of utilities under the control of the operational environment.
A.NO_BYPASS	The operational environment will ensure the TSF cannot be bypassed in order to gain access to TOE data.
A.NO_EVIL	Administrators are non-hostile, appropriately trained, and follow all administrator guidance.
A.NO_GENERAL_PURPOSE	There are no general-purpose computing capabilities (e.g., compilers or user applications) available on the DBMS servers, other than those services necessary for the operation, administration, and support of the DBMS.
A.PHYSICAL	It is assumed that appropriate physical security is provided within the domain for the value of the IT assets protected by the TOE and the value of the stored, processed, and transmitted information.
A.RESTRICT_OS_ACCESS	Logon access to the underlying operating system is restricted to authorized administrators only.
A.ROBUST_ENVIRONMENT	The operational environment is at least as robust as the TOE.
A.SECURE_COMMS	The operational environment will provide a secure (protected from disclosure, spoofing, and able to detect modification) line of communications between the remote user and the TOE.
A.TIME_STAMPS	The operational environment will provide the TOE with the necessary reliable timestamps.

4 Security Objectives

Security objectives are concise, abstract statements of the intended solution to the problem defined by the security problem definition (see Section 3). The set of security objectives for a TOE form a high-level solution to the security problem. This high-level solution is divided into two part-wise solutions: the security objectives for the TOE, and the security objectives for the TOE's operational environment. This section identifies the security objectives for the TOE and its supporting environment.

4.1 Security Objectives for the TOE

The specific security objectives for the TOE are listed in Table 7 below.

Table 7 – Security Objectives for the TOE

Name	Description
O.ACCESS_HISTORY	The TOE will store and retrieve information (to authorized users) related to previous attempts to establish a session.
O.ADMIN_GUIDANCE	The TOE will provide administrators with the necessary information for secure management.
O.ADMIN_ROLE	The TOE will provide authorized administrator roles to isolated administrative actions.
O.AUDIT_GENERATION	The TOE will provide the capability to detect and create records of security relevant events associated with users.
O.AUDIT_REVIEW	The TOE will contain mechanisms to allow the authorized security administrator to view and sort the audit logs.
O.AUDIT_STORAGE	The TOE will contain mechanisms to provide secure storage and management of the audit log.
O.CONFIGURATION_IDENTIFICATION	The configuration of the TOE is fully identified in a manner that will allow implementation errors to be identified and corrected with the TOE being redistributed promptly.
O.DOCUMENTED_DESIGN	The design of the TOE is adequately and accurately documented.
O.FAIL_SECURE	The TOE will provide mechanisms to allow for secure failure and recovery.
O.I&A	The TOE will contain identification and authentication mechanisms for users to login to the TOE.
O.INTERNAL_TOE_DOMAINS	The TSF will maintain internal domains for separation of data and queries belonging to concurrent users.
O.MANAGE	The TOE will provide all the functions and facilities necessary to support the authorized security administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.
O.MEDIATE	The TOE must protect user data in accordance with its security policy.
O.PARTIAL_FUNCTIONAL_TEST	The TOE will undergo some security functional testing that demonstrates that the TSF satisfies some of its security functional

Name	Description
	requirements.
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.QUOTAS	The TOE will provide the ability to define quotas for usage of TOE resources, and prevent further allocation of resources once the quotas are reached.
O.RESIDUAL_INFORMATION	The TOE will ensure that any information contained within its Scope of Control is not released when the resource is reallocated.
O.TOE_ACCESS	The TOE will provide mechanisms that control a user's logical access to the TOE.
O.VULNERABILITY_ANALYSIS	The TOE will undergo some vulnerability analysis to demonstrate that the design and implementation of the TOE does not contain any obvious flaws.

4.2 Security Objectives for the Operational Environment

This section describes the environmental objectives.

4.2.1 IT Security Objectives

Table 8 below lists the IT security objectives that are to be satisfied by the environment.

Table 8 – IT Security Objectives

Name	Description
OE.DOMAIN_SEPARATION	The operational environment will provide an isolated domain for the execution of the TOE.
OE.I&A	The operational environment will contain identification and authentication mechanisms for administrator access to database control utilities and other utilities.
OE.NO_BYPASS	The operational environment shall ensure the TOE security mechanisms cannot be bypassed in order to gain access to the TOE resources.
OE.NO_GENERAL_PURPOSE	There will be no general-purpose computing capabilities (e.g., compilers or user applications) available on DBMS servers, other than those services necessary for the operation, administration and support of the DBMS.
OE.RESTRICT_OS_ACCESS	The underlying operating system will be configured with only those user accounts required for access by authorized security administrators.
OE.ROBUST_ENVIRONMENT	The operational environment that supports the TOE for enforcement of its security objectives will be of at least the same level of

Name	Description
	robustness as the TOE.
OE.SECURE_COMMS	The operational environment will provide a secure line of communications between the remote user and the TOE.
OE.TIME_STAMPS	The operational environment will provide reliable time stamps.
OE.TRUST_IT	Each operational entity the TOE relies on for security functions will be installed, configured, managed and maintained in a manner appropriate to the IT entity, and consistent with the security policy of the TOE and the relationship between them.

4.2.2 Non-IT Security Objectives

Table 9 below lists the non-IT environment security objectives that are to be satisfied without imposing technical requirements on the TOE. That is, they will not require the implementation of functions in the TOE hardware and/or software. Thus, they will be satisfied largely through application of procedural or administrative measures.

Table 9 – Non-IT Security Objectives

Name	Description
NOE.NO_EVIL	Sites using the TOE shall ensure that authorized administrators are non-hostile, appropriately trained, and follow all administrator guidance.
NOE.PHYSICAL	Physical security will be provided within the domain for the value of the IT assets protected by the TOE and the value of the stored, processed, and transmitted information.

5 Extended Components

This section defines the extended SFRs and extended SARs met by the TOE. These requirements are presented following the conventions identified in Section 6.1.

5.1 Extended TOE Security Functional Components

The Extended TOE SFRs are defined in the DBMS PP. No additional extended SFRs are defined for this ST.

5.1.1 Class FAU: Security Audit

Families in this class address the requirements for generation of security audit data as defined in CC Part 2.

5.1.1.1 Family FAU_GEN_(EXT): User and/or group identity association

Family Behaviour

This family defines requirements for recording the occurrence of security relevant events that take place under TSF control. This family identifies the level of auditing, enumerates the types of events that shall be auditable by the TSF, and identifies the minimum set of audit-related information that should be provided within various audit record types.

Components in this family address the requirements for audit data as defined in CC Part 2. This section defines the extended components for the FAU_GEN family.

Component leveling

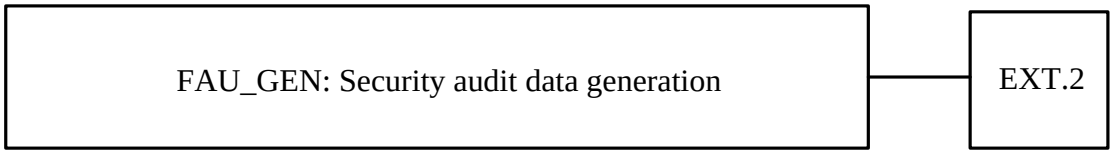


Figure 3 – Security audit data generation family decomposition

The extended FAU_GEN_(EXT).2 component is considered to be part of the FAU_GEN family.

FAU_GEN_(EXT).2 Extended: User and/or group identity association specifies the audit function will associate a user identity or a group identity or both with all events resulting from the action of a user as defined in the DBMS PP. This SFR was modeled after FAU_GEN.2.

Management: FAU_GEN_(EXT).2

- There are no management activities foreseen.

Audit: FAU_GEN_(EXT).2

- There are no auditable events foreseen.

FAU_GEN_(EXT).2 User and/or group identity association
Hierarchical to: No other components.
FAU_GEN_(EXT).2.1

For audit events resulting from actions of identified users and/or identified groups, the TSF shall be able to associate each auditable event with the identity of the user and/or group that caused the event.

Dependencies: **FAU_GEN.1 Audit data generation**
FIA_UID.1 Timing of identification

5.1.2 Class FMT: Security Management

Families in this class specify the management of several aspects of the TSF as defined in CC Part 2.

5.1.2.1 Family FMT_MSA: Management of security attributes

Family Behaviour

This family allows authorised users control over the management of security attributes. This management might include capabilities for viewing and modifying of security attributes.

This section defines the extended components for the FMT_MSA family.

Component Leveling

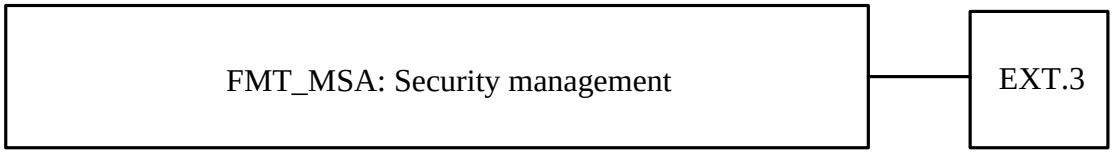


Figure 4 – Security management family decomposition

The extended FMT_MSA_(EXT).3 component is considered to be part of the FMT_MSA family as defined in CC Part 2.

FMT_MSA_(EXT).3 Extended: This extended requirement requires the security attributes of the objects on creation to be restrictive and not allowing any user to be able to override the restrictive default values as defined in the DBMS PP. This SFR was modeled after FMT_MSA.3.

Management FMT_MSA_(EXT).3

The following actions could be considered for the management functions in FMT:

- managing the restrictive setting of default values for a given access control SFP;
- management of rules by which security attributes inherit specified values.

Audit FMT_MSA_(EXT).3

There are no auditable events foreseen.

FMT_MSA_(EXT).3 Static attribute initialisation
Hierarchical to: No other components.
FMT_MSA_(EXT).3.1
The TSF shall enforce the [Discretionary Access Control policy] to provide *restrictive* default values for security attributes that are used to enforce the SFP.
Dependencies: No Dependencies

5.1.3 Class FPT: Protection of the TSF

The families in this class relate to the integrity and management of the mechanisms that constitute the TSF and to the integrity of TSF data as defined in CC Part 2.

5.1.3.1 Family FPT_TRC: Internal TOE TSF data replication consistency

Family Behaviour

The requirements of this family are needed to ensure the consistency of TSF data when such data is replicated internal to the TOE. Such data may become inconsistent if the internal channel between parts of the TOE becomes inoperative. If the TOE is internally structured as a network and parts of the TOE network connections are broken, this may occur when parts become disabled.

This section defines the extended components for the FPT_TRC family.

Component Leveling

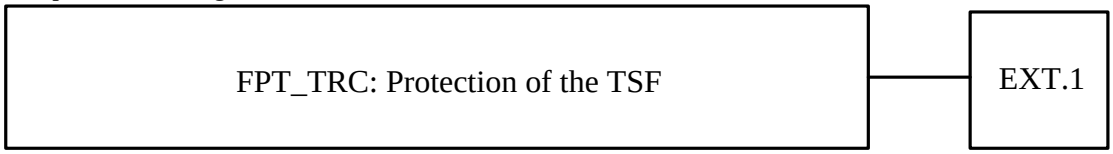


Figure 5 – Protection of the TSF family decomposition

The extended FPT_TRC_(EXT).1 component is considered to be part of the FPT_TRC family as defined in CC Part 2.

FPT_TRC_(EXT).1 Extended: Requires timely consistency of replicated TSF data as defined in DBMS PP. This SFR was modeled after FPT_TRC.1.

Management: FPT_TRC_(EXT).1

There are no management activities foreseen.

Audit: FPT_TRC_(EXT).1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

- Minimal: restoring consistency after shortly after reconnection
- Basic: detected inconsistency between TSF data

FPT_TRC_(EXT).1 Internal TSF consistency
Hierarchical to: No other components.
FPT_TRC_(EXT).1.1
The TSF shall ensure that TSF data is consistent between parts of the TOE by providing a mechanism to bring inconsistent TSF data into a consistent state in a timely manner.
Dependencies: No dependencies

5.1.4 Class FTA: TOE Access

Families in this class address the requirements for functions that control the establishment and existence of a user session as defined in CC Part 2.

5.1.4.1 Family FTA_TAH: TOE access history

This family defines requirements for the TSF to display to a user, upon successful session establishment, a history of successful and unsuccessful attempts to access the user's account.

The extended FTA_TAH_(EXT).1 component is considered to be part of the FTA_TAH family.

Component Leveling

Figure 6 – TOE access history family decomposition

FTA_TAH_(EXT).1 Extended: Requires the TOE to store and retrieve the access history as defined in DBMS PP. This SFR is modeled after FTA_TAH.1.

Management: FTA_TAH_(EXT).1

There are no management activities foreseen.

Audit: FTA_TAH_(EXT).1

There are no audit activities foreseen.

FTA_TAH_(EXT).1 TOE access history
Hierarchical to: No other components.
FTA_TAH_(EXT).1.1
Upon successful session establishment, the TSF shall store and retrieve the *date and time* of the last successful session establishment to the user.
FTA_TAH_(EXT).1.2
Upon successful session establishment, the TSF shall store and retrieve the *date and time* of the last unsuccessful attempt to session establishment and the number of unsuccessful attempts since the last successful session establishment.
Dependencies: No dependencies

5.2 Extended TOE Security Assurance Components

This Security Target does not define any extended assurance components.

6 Security Requirements

This section defines the SFRs and SARs met by the TOE. These requirements are presented following the conventions identified in Section 6.1.

6.1 Conventions

There are several font variations used within this ST. Selected presentation choices are discussed here to aid the Security Target reader.

The CC allows for assignment, refinement, selection and iteration operations to be performed on security functional requirements. All of these operations are used within this ST. These operations are performed as described in Part 2 of the CC, and are shown as follows:

- Completed assignment statements are identified using [*italicized text within brackets*].
- Completed selection statements are identified using [underlined text within brackets].
- Refinements are identified using **bold text**. Any text removed is stricken (Example: ~~TSF Data~~) and should be considered as a refinement.
- Extended Functional and Assurance Requirements are identified using "EXT_" at the beginning of the short name. If an extended requirement was defined within the DBMS PP, then the name given in the PP is used instead.
- Iterations are identified by appending a number in parentheses following the component title. For example, FAU_GEN.1(1) Audit Data Generation would be the first iteration and FAU_GEN.1(2) Audit Data Generation would be the second iteration.

Many SFRs are taken from the DBMS PP and use terms from the DBMS PP. These terms are defined in Section 9.2.

6.2 Security Functional Requirements

This section specifies the SFRs for the TOE. This section organizes the SFRs by CC class. Table 10 identifies all SFRs implemented by the TOE and indicates the ST operations performed on each requirement.

Table 10 – TOE Security Functional Requirements

Name	Description	S	A	R	I
FAU_GEN.1-NIAP-0410	Audit Data Generation	✓	✓	✓	
FAU_GEN_(EXT).2	User and/or group identity association				
FAU_SAR.1	Audit review		✓		
FAU_SAR.2	Restricted audit review				
FAU_SAR.3	Selectable audit review		✓		
FAU_SEL.1-NIAP-0407	Selective audit	✓	✓	✓	
FAU_STG.1	Protected audit trail storage	✓			
FDP_ACC.1	Subset access control		✓		
FDP_ACF.1-NIAP-0407	Security attribute based access control		✓	✓	
FDP_RIP.1	Subset residual information protection	✓	✓		
FIA_ATD.1	User attribute definition		✓		

Name	Description	S	A	R	I
FIA_UAU.2	User authentication before any action				
FIA_UID.2	User identification before any action				
FIA_USB.1	User-subject binding		✓		
FMT_MOF.1	Management of security functions behaviour	✓	✓		
FMT_MSA.1	Management of security attributes	✓	✓	✓	
FMT_MSA_(EXT).3	Static attribute initialisation	✓	✓		
FMT_MTD.1	Management of TSF data		✓		
FMT_REV.1(1)	Revocation		✓		
FMT_REV.1(2)	Revocation		✓		
FMT_SMF.1	Specification of management functions		✓		
FMT_SMR.1	Security roles		✓	✓	
FPT_FLS.1	Failure with preservation of secure state		✓		
FPT_TRC_(EXT).1	Internal TSF consistency				
FRU_FLT.1	Degraded fault tolerance		✓		
FRU_RSA.1	Maximum quotas	✓	✓		
FTA_MCS.1	Basic limitation on multiple concurrent sessions		✓	✓	
FTA_TAH_(EXT).1	TOE access history				
FTA_TSE.1	TOE session establishment		✓	✓	

Note: S=Selection; A=Assignment; R=Refinement; I=Iteration

6.2.1 Class FAU: Security Audit

FAU_GEN.1-NIAP-0410 Audit Data Generation
Hierarchical to: No other components.
FAU_GEN.1.1-NIAP-0410

Refinement: 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 *minimum* level of audit **listed in Table 11**;
c) **[Start-up and shutdown of the DBMS;**
d) **Use of special permissions (e.g., those often used by authorized administrators[9] to circumvent access control policies); and**
e) [[*the auditable events listed in Table 11 below, segment database failures, and SQL statements*]].

FAU_GEN.1.2-NIAP-0410

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 (if applicable), 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, [*information specified in column three of Table 11 below, session number, process and thread ID[10], segment name, database name, IP address, transaction ID, and severity*].

Dependencies: FPT_STM.1 Reliable time stamps

Table 11 – Auditable Events

Security Functional Requirement	Auditable Event(s)	Additional Audit Record Contents
FAU_GEN.1-NIAP-0410	None	None
FAU_GEN_(EXT).2	None	None
FAU_SAR.1	None	None
FAU_SAR.2	None	None
FAU_SAR.3	None	None
FAU_STG.1	None	None
FAU_SEL.1-NIAP-0407	All modifications to the audit configuration that occur while the audit collection functions are operating	The identity of the authorized administrator that made the change to the audit configuration
FDP_ACC.1	None	None
FDP_ACF.1-NIAP-0407	Successful requests to perform an operation on an object covered by the SFP	The identity of the subject performing the operation
FIA_ATD.1	None	None
FIA_UAU.2	Unsuccessful use of the authentication mechanism.	None

[9] Authorized administrator refers to the SuperUser administrative account, or any administrative account with permissions (set by the SuperUser) that allow access to the function.
[11] The designation "Discretionary Access Control policy" has been changed here for the sake of clarity.

Security Functional Requirement	Auditable Event(s)	Additional Audit Record Contents
FIA_UID.2	Unsuccessful use of the identification mechanism, including the user identity provided	None
FIA_USB.1	Unsuccessful binding of user security attributes to a subject (e.g. creation of a subject)	None
FMT_MOF.1	None	None
FMT_MSA.1	None	None
FMT_MSA_(EXT).3	None	None
FMT_MTD.1	None	None
FMT_REV.1(1)	Unsuccessful revocation of security attributes	Identity of individual attempting to revoke security attributes
FMT_REV.1(2)	Unsuccessful revocation of security attributes	Identity of individual attempting to revoke security attributes
FMT_SMF.1	Use of management functions	Identity of the administrator performing these functions
FMT_SMR.1	Modifications to the group of users that are part of a role	Identity of authorized administrator modifying the role definition
FPT_FLS.1	None	None
FPT_TRC_(EXT).1	Restoring consistency	None
FRU_FLT.1	Any failure detected by the TSF	None
FRU_RSA.1	Rejection of allocation operation due to resource limits.	Identity of the subject requesting resources
FTA_MCS.1	Rejection of a new session based on the limitation of multiple concurrent sessions	None
FTA_TAH_(EXT).1	None	None
FTA_TSE.1	Denial of a session establishment due to the session establishment mechanism	Identity of the individual attempting to establish a session

FAU_GEN_(EXT).2 User and/or group identity association

Hierarchical to: No other components.

FAU_GEN_(EXT).2.1

For audit events resulting from actions of identified users and/or identified groups, the TSF shall be able to associate each auditable event with the identity of the user and/or group that caused the event.

Dependencies: FAU_GEN.1 Audit data generation
FIA_UID.1 Timing of identification

Application Note: The database username is used in all audit records to uniquely identify the user or group. By default the database username and role name are the same.

FAU_SAR.1 Audit review

Hierarchical to: No other components.

FAU_SAR.1.1

The TSF shall provide [*authorised roles*] 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

FAU_SAR.2 Restricted audit review

Hierarchical to: No other components.

FAU_SAR.2.1

The TSF shall prohibit all users read access to the audit records, except those users that have been granted explicit read-access.

Dependencies: FAU_SAR.1 Audit review

FAU_SAR.3 Selectable audit review

Hierarchical to: No other components.

FAU_SAR.3.1

The TSF shall provide the ability to apply [*SQL querying*] of audit data based on [*valid criteria for SELECT, WHERE, DISTINCT, ORDER BY, GROUP BY, HAVING, AND, OR, COUNT, AVG, MIN MAX, and SUM SQL commands, and any valid combination thereof*].

Dependencies: FAU_SAR.1 Audit review

FAU_SEL.1-NIAP-0407 Selective audit

Hierarchical to: No other components.

FAU_SEL.1.1

The TSF shall ~~be able to~~ **allow only the administrator to include or exclude auditable events from the set of audited** events based on the following attributes:

a) ~~user identity and/or group identity~~,
b) event type,
c) ~~object identity~~,
d) [none]
e) [success of auditable security events;
f) failure of auditable security events; and
g) [[*no additional criteria*].]

Dependencies: FAU_GEN.1 Audit data generation
FMT_MTD.1 Management of TSF data

Application Note: User identity and /or group identity and object identity have been stricken out of the SFR because they are not applicable to the TOE. This SFR remains in conformance with the PP based on application note 70 in the PP. The TOE audit function allows administrators to capture enough data to perform their tasks. The level of auditing does not consume more resources than the TOE can handle and audit records can be sorted by user identity or object identity.

FAU_STG.1 Protected audit trail storage

Hierarchical to: No other components.

FAU_STG.1.1

The TSF shall protect the stored audit records in the audit trail from unauthorised deletion.

FAU_STG.1.2

The TSF shall be able to [prevent] unauthorised modifications to the stored audit records in the audit trail.

Dependencies: FAU_GEN.1 Audit data generation

6.2.2 Class FDP: User Data Protection

FDP_ACC.1 Subset access control
Hierarchical to: No other components.
FDP_ACC.1.1
The TSF shall enforce the [*Data Access SFP*[11]]] on [*all subjects, all DBMS-controlled objects and all operations among them*].
Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACF.1-NIAP-0407 Security attribute based access control
Hierarchical to: No other components.
FDP_ACF.1.1-NIAP-0407
The TSF shall enforce the [*Data Access SFP*] to objects based on the following:
- [the authorized user identity and/or group membership associated with a subject;
- access operations implemented for **DBMS**-controlled objects; and
- object identity].

FDP_ACF.1.2-NIAP-0407
The TSF shall enforce the following rules to determine if an operation among controlled subjects and DBMS-controlled objects is allowed:

The Data Access SFP mechanism shall, either by explicit authorized user/group action or by default, provide that database management system controlled objects are protected from unauthorized access according to the following ordered rules:

[
*a) If the requested mode of access is denied to that authorized user **ID**, deny access;*
*b) If the requested mode of access is permitted to that authorized user **ID**, permit access;*
*c) If the requested mode of access is denied to every group **ID** of which the authorized user is a member, deny access;*
*d) If the requested mode of access is permitted to any group **ID** of which the authorized user is a member, grant access;*
e) Else, deny access
].

FDP_ACF.1.3-NIAP-0407
The TSF shall explicitly authorize access of subjects to **DBMS-controlled** objects based on the following additional rules: [*an authorized administrator has granted access to run a SQL statement or MapReduce command on the data*].

FDP_ACF.1.4-NIAP-0407
The TSF shall explicitly deny access of subjects to objects based on the following rules: [*an authorized administrator has revoked access to run a SQL statement or MapReduce command on the data*].

Dependencies: FDP_ACC.1 Subset access control
FMT_MSA.3 Static attribute initialization

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 list of objects in column 1 Table 16*].
Dependencies: No dependencies
Application Note: The TOE requests cleared memory which is provided by the TOE environment (the OS). RHEL v5.5, 5.7, and 6.1 are capable of providing the cleared memory as requested by the TOE.

[11] The designation "Discretionary Access Control policy" has been changed here for the sake of clarity.

6.2.3 Class FIA: Identification and Authentication

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:

- [*Database user identifier and/or group memberships;*
- *Security-relevant database roles; and*
- [*authentication type and password*]].

Dependencies: No dependencies

FIA_UAU.2 User authentication before any action
Hierarchical to: FIA_UAU.1.
FIA_UAU.2.1
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

FIA_UID.2 User identification before any action
Hierarchical to: FIA_UID.1.
FIA_UID.2.1
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

FIA_USB.1: User-subject binding
Hierarchical to: No other components
FIA_USB.1.1:
The TSF shall associate the following user security attributes with subjects acting on the behalf of that user: [*authenticated user roles*].
FIA_USB.1.2:
The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users: [*the user must authenticate successfully with the TOE before any associations are made*].
FIA_USB.1.3:
The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users: [*any changes to user roles take effect immediately after the additional privileges are granted*].
Dependencies: FIA_ATD.1 User Attribute Definition

6.2.4 Class FMT: Security Management

FMT_MOF.1 Management of security functions behaviour
Hierarchical to: No other components.
FMT_MOF.1.1
The TSF shall restrict the ability to [*disable and enable*] the functions [*relating to the specification of events to be audited*] to [*authorized administrators*].
Dependencies: FMT_SMF.1 Specification of management functions
FMT_SMR.1 Security roles

FMT_MSA.1 Management of security attributes
Hierarchical to: No other components.
FMT_MSA.1.1
The TSF shall enforce the [*Data Access SFP*] to restrict the ability to [*manage*] **all** the security attributes to [*authorized administrators*].
Dependencies: FDP_ACC.1 Subset access control
FMT_SMF.1 Specification of management functions
FMT_SMR.1 Security roles

FMT_MSA_(EXT).3 Static attribute initialisation
Hierarchical to: No other components.
FMT_MSA_(EXT).3.1
The TSF shall enforce the [*Data Access SFP*] to provide *restrictive* default values for security attributes that are used to enforce the SFP.
Dependencies: No Dependencies

FMT_MTD.1 Management of TSF data
Hierarchical to: No other components.
FMT_MTD.1.1
The TSF shall restrict the ability to [*include or exclude*] the [*auditable events*] to [*authorized administrators*].
Dependencies: FMT_SMF.1 Specification of management functions
FMT_SMR.1 Security roles

FMT_REV.1(1) Revocation
Hierarchical to: No other components.
FMT_REV.1.1(1)
The TSF shall restrict the ability to revoke security attributes associated with the *users* within the TSC to [*the authorized administrator*].
FMT_REV.1.2(1)
The TSF shall enforce the rules [*no additional rules*].
Dependencies: FMT_SMR.1 Security roles

FMT_REV.1(2) Revocation
Hierarchical to: No other components.
FMT_REV.1.1(2)
The TSF shall restrict the ability to revoke security attributes associated with the *objects* within the TSC to [*the authorized administrator and database users as allowed by the Data Access SFP*].
FMT_REV.1.2(2)
The TSF shall enforce the rules [*no additional rules*].
Dependencies: FMT_SMR.1 Security roles

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 management functions: [*management of security attributes, management of TSF data, management of security functions behavior*].

Dependencies: No Dependencies

FMT_SMR.1 Security roles

Hierarchical to: No other components.

FMT_SMR.1.1

Refinement: The TSF shall maintain the roles

- [authorized administrator]; **and**
- **[*roles with custom permissions, Object Owners*]**.

FMT_SMR.1.2

The TSF shall be able to associate users with roles.

Dependencies: FIA_UID.1 Timing of identification

6.2.5 Class FPT: Protection of the TSF

FPT_FLS.1 Failure with preservation of secure state
Hierarchical to: No other components.
FPT_FLS.1.1
The TSF shall preserve a secure state when the following types of failures occur: [*failure of the master node*].
Dependencies: No dependencies.

FPT_TRC_(EXT).1 Internal TSF consistency
Hierarchical to: No other components.
FPT_TRC_(EXT).1.1
The TSF shall ensure that TSF data is consistent between parts of the TOE by providing a mechanism to bring inconsistent TSF data into a consistent state in a timely manner.
Dependencies: No dependencies.

6.2.6 Class FRU: Resource Utilization

FRU_FLT.1 Degraded fault tolerance
Hierarchical to: No other components.
FRU_FLT.1.1
The TSF shall ensure the operation of [*segments and the back-end private network*] when the following failures occur: [*failure of a master*].
Dependencies: FPT_FLS.1 Failure with preservation of secure state

FRU_RSA.1 Maximum quotas
Hierarchical to: No other components.
FRU_RSA.1.1
The TSF shall enforce maximum quotas of the following resources: [*processing capabilities*] that [individual user] can use [simultaneously].
Dependencies: No dependencies

6.2.7 Class FTA: TOE Access

FTA_MCS.1 Basic limitation on multiple concurrent sessions
Hierarchical to: No other components.
FTA_MCS.1.1
The TSF shall restrict the maximum number of concurrent sessions that belong to the same user **except SuperUsers**.
FTA_MCS.1.2
The TSF shall enforce, by default, a limit of [*no limit on*] sessions per user.
Dependencies: FIA_UID.1 Timing of identification

FTA_TAH_(EXT).1 TOE access history
Hierarchical to: No other components.
FTA_TAH_(EXT).1.1
Upon successful session establishment, the TSF shall store and retrieve the *date and time* of the last successful session establishment to the user.
FTA_TAH_(EXT).1.2
Upon successful session establishment, the TSF shall store and retrieve the *date and time* of the last unsuccessful attempt to session establishment and the number of unsuccessful attempts since the last successful session establishment.
Dependencies: No dependencies

FTA_TSE.1 TOE session establishment
Hierarchical to: No other components.
FTA_TSE.1.1
Refinement: The TSF shall be able to deny session establishment based on [attributes that can be set explicitly by authorized administrator(s), including user identity and/or group identity, time of day, day of the week], **and [*no additional attributes*]**.
Dependencies: No dependencies

6.3 Security Assurance Requirements

This section defines the assurance requirements for the TOE. Assurance requirements are taken from the CC Part 3 and are EAL2 augmented with ALC_FLR.2. Table 12 – Assurance Requirements summarizes the requirements.

Table 12 – Assurance Requirements

Assurance Requirements	
Class ASE: Security Target evaluation	ASE_CCL.1 Conformance claims
	ASE_ECD.1 Extended components definition
	ASE_INT.1 ST introduction
	ASE_OBJ.2 Security objectives
	ASE_REQ.2 Derived security requirements
	ASE_SPD.1 Security problem definition
	ASE_TSS.1 TOE summary specification
Class ALC : Life Cycle Support	ALC_CMC.2 Use of a CM system
	ALC_CMS.2 Parts of the TOE CM coverage
	ALC_DEL.1 Delivery Procedures
	ALC_FLR.2 Flaw reporting procedures
Class ADV: Development	ADV_ARC.1 Security architecture description
	ADV_FSP.2 Security-enforcing Functional Specification
	ADV_TDS.1 Basic design
Class AGD: Guidance documents	AGD_OPE.1 Operational user guidance
	AGD_PRE.1 Preparative procedures
Class ATE: Tests	ATE_COV.1 Evidence of coverage
	ATE_FUN.1 Functional testing
	ATE_IND.2 Independent testing – sample
Class AVA: Vulnerability assessment	AVA_VAN.2 Vulnerability analysis

7 TOE Summary Specification

This section presents information to detail how the TOE meets the functional requirements described in previous sections of this ST.

7.1 TOE Security Functions

Each of the security requirements and the associated descriptions correspond to the security functions. Hence, each function is described by how it specifically satisfies each of its related requirements. This serves to both describe the security functions and rationalize that the security functions satisfy the necessary requirements. Table 13 lists the security functions and their associated SFRs.

Table 13 – Mapping of TOE Security Functions to Security Functional Requirements

TOE Security Function	SFR ID	Description
Security Audit	FAU_GEN.1-NIAP-0410	Audit Data Generation
	FAU_GEN_(EXT).2	User and/or group identity association
	FAU_SAR.1	Audit review
	FAU_SAR.2	Restricted audit review
	FAU_SAR.3	Selectable audit review
	FAU_SEL.1-NIAP-0407	Selective audit
	FAU_STG.1	Protected audit trail storage
User Data Protection	FDP_ACC.1	Subset access control
	FDP_ACF.1-NIAP-0407	Security attribute based access control
	FDP_RIP.1	Subset residual information protection
Identification and Authentication	FIA_ATD.1	User attribute definition
	FIA_UAU.2	User authentication before any action
	FIA_UID.2	User identification before any action
	FIA_USB.1	User-subject binding
Security Management	FMT_MOF.1	Management of security functions behaviour
	FMT_MSA.1	Management of security attributes
	FMT_MSA_(EXT).3	Static attribute initialisation
	FMT_MTD.1	Management of TSF data
	FMT_REV.1(1)	Revocation
	FMT_REV.1(2)	Revocation
	FMT_SMF.1	Specification of management

TOE Security Function	SFR ID	Description
		functions
	FMT_SMR.1	Security roles
Protection of the TSF	FPT_FLS.1	Failure with preservation of secure state
	FPT_TRC_(EXT).1	Internal TSF consistency
Resource Utilization	FRU_FLT.1	Degraded fault tolerance
	FRU_RSA.1	Maximum quotas
TOE Access	FTA_MCS.1	Basic limitation on multiple concurrent sessions
	FTA_TAH_(EXT).1	TOE access history
	FTA_TSE.1	TOE session establishment

7.1.1 Security Audit

The TOE is capable of auditing a variety of events, including startup and shutdown of the system, segment database failures, SQL statements that result in an error, and all connection attempts and disconnects (if configured). The TOE also logs SQL statements and information regarding SQL statements, and can be configured in a variety of ways to record audit information with more or less detail.[12] For example, the log_error_verbosity configuration parameter controls the amount of detail written in the server log for each message that is logged. Similarly, the log_min_error_statement parameter allows administrators to configure the level of detail recorded specifically for SQL statements, and the log_statement determines the kind of SQL statements audited. The TOE records the database username for all auditable events, when the event was initiated by a subject outside the TOE.

Although the TOE does not explicitly generate an audit event for startup and shutdown of the audit functionality, it does audit the startup and shutdown of the system. Since the system starts up and shuts down at the same time as the audit function, these records can be considered to provide equivalent notice.

The TOE has an extensive set of auditing features, including the ability to review logs by using the "external table" facility. This facility allows users to access data in external sources (such as the log files) as though it were data in a database, meaning that it can be sorted and ordered with regular SQL commands. Additionally, administrators can view the logs via Command Line Interface (CLI) commands and scripts. These scripts, combined with the gplogfilter utility or the "cat" and "grep" commands (for concatenation and keyword selection), can be used to filter log files. Access to view logs is based on each administrator's role. Administrators can also include or exclude events from the list of audited events based upon event type or success or failure of the event. This selectivity of the audit function allows administrators to capture enough data to perform their tasks. The TOE is capable of processing 1 – 2 GB of log files per second and can store terabytes of log files, so this level of auditing does not consume more resources than the TOE can handle. The audit records can be sorted by user identity, role identity, or object identity.

The TOE prevents unauthorized modification and deletion of audit records by only allowing administrators with a sufficient role to perform any operations on log files. The TOE also authorizes overwriting of old log files, if configured to do so, via the truncate_on_rotation parameter.

[12] The TOE also generates audit records to cover the security-relevant events listed in Table 11.

Logs are stored in a proprietary format using Comma-Separated Values (CSV). Each segment and the master store their own log files, although these can be accessed remotely by an administrator. Table 14 below lists the information that TOE audit records contain[13].

Table 14 – Audit Record Contents

Field	Content
event_time	Time stamp with time zone that the log entry was written to the log.
user_name	The database user name.
database_name	The database name.
process_id	The system process id (prefixed with "p").
thread_id	The thread count (prefixed with "th").
remote_host	On the master: the client machine hostname and address. On the segment: the hostname and address of the master.
remote_port	On the master: the client machine port. On the segment: the port of the master.
session_start_time	Time stamp with time zone that the session connection was opened.
transaction_id	Top-level transaction ID on the master. This is the parent of any subtransactions.
gp_session_id	Session identifier number (prefixed with "con").
gp_command_count	The command number within a session (prefixed "cmd").
gp_segment	The segment containing the identifier (prefixed with "seg" for primaries or "mir" for mirrors). The master always has a content id of -1.
slice_id	The slice id (portion of the query plan being executed).
distr_tranx_id	Distributed transaction ID.
local_tranx_id	Local transaction ID.
sub_tranx_id	Subtransaction ID.
event_severity	Values include: LOG, ERROR, FATAL, PANIC, DEBUG1, and DEBUG2.
sql_state_code	SQL state code associated with the log message.
event_message	Log or error message text.
event_detail	Detail message text associated with an error or warning message.
event_hint	Hint message text associated with an error or warning message.

[13] Please note that not all of these fields will appear within every log message.

Field	Content
internal_query	The internally-generated query text.
internal_query_pos	The cursor index into the internally-generated query text.
event_context	The context in which this message gets generated.
debug_query_string	User-supplied query string with full detail for debugging. Greenplum DBMS may modify this string for internal use.
error_cursor_pos	The cursor index into the query string.
func_name	The function in which this message is generated.
file_name	The internal code file where the message originated.
file_line	The line of the code file where the message originated.
stack_trace	Stack trace text associated with this message.

TOE Security Functional Requirements Satisfied: FAU_GEN.1-NIAP-0410, FAU_GEN_(EXT).2, FAU_SAR.1, FAU_SAR.2, FAU_SAR.3, FAU_SEL.1-NIAP-0407, FAU_STG.1.

7.1.2 User Data Protection

The TOE provides access control mechanisms that prevent unauthorized access to data. Access is assigned via roles to individual users, groups of users, both, or even other roles. Roles can own database objects, such as tables, and control the access permissions of other roles to those objects. Roles that are members of another role inherit the privileges of the parent role.

Users log in by authenticating with the master. After successful authentication, the master assumes the user's roles when distributing jobs to the various segments. Roles are valid for all databases controlled by the TOE. Table 15 below lists the attributes that a role can have.

Table 15 – Role Attributes

Attribute	Description
SUPERUSER | NOSUPERUSER	Determines if the role is a superuser.
CREATEDB | NOCREATEDB	Determines if the role is allowed to create databases.
CREATEROLE | NOCREATEROLE	Determines if the role is allowed to create and manage other roles.
INHERIT | NOINHERIT	Determines whether a role inherits the privileges of roles it is a member of.
LOGIN | NOLOGIN	Determines whether a role is allowed to log in. Typically a role with NOLOGIN set is used as a group role.
CONNECTION LIMIT *connlimit*	If role can log in, this specifies how many concurrent sessions the role can make.
PASSWORD *'password'*	Sets the role's password.
VALID UNTIL *'timestamp'*	Sets a date and time after which the role's password is no longer valid.

Attribute	Description
RESOURCE QUEUE *queue_name*	Assigns the role to the named resource queue for workload management. Any statement that role issues is then subject to the resource queue's limits. This attribute is not inherited.

In addition to role attributes, the TOE also controls the object privileges listed in Table 16 below.

Table 16 – Object Privileges

Object Type	Privileges
Tables, Views, Sequences	SELECT, INSERT, UPDATE, DELETE, RULE, ALL
External Tables	SELECT, RULE, ALL
Databases	CONNECT, CREATE, TEMPORARY | TEMP, ALL
Functions	EXECUTE
Procedural Languages	USAGE
Schemas	CREATE, USAGE, ALL

The TOE's process-based architecture works with the OS in the TOE environment to ensure that residual information protection occurs. At session initiation a new, unique process or set of processes is created and linked to a new allocation of memory. The TOE requests cleared memory allocation from the operating environment ensuring that memory allocated to a new process is zeroized.
TOE Security Functional Requirements Satisfied: FDP_ACC.1, FDP_ACF.1.

7.1.3 Identification and Authentication

The TOE requires users and administrators to authenticate with the TOE before being presented with any TOE functionality. Users and administrators can use passwords to authenticate with the TOE. The TOE stores the credentials associated with each user and administrator locally, including the user ID, authentication type, password, and any roles applied to the user's or administrator's account.

When users or administrators authenticate with the TOE via the master, the master assumes users' or administrators' roles to send commands on behalf of the user to the segments. In this way, users bind to the master rather than issuing commands directly.
TOE Security Functional Requirements Satisfied: FIA_ATD.1, FIA_UAU.2, FIA_UID.2, FIA_USB.1.

7.1.4 Security Management

Management of the TOE is performed from a third-party database client application that is not part of the TOE. These applications are designed to connect to a database server and provide only an engine to pass commands and view database objects. Database client applications can present command line or graphical interfaces. Since all configuration data is stored internally by the database, these applications are used for both user access and administrative access to the TOE.

The TOE can be configured by modifying a set of master and local parameters. The master and each segment have their own configuration file that stores all of the local parameters. Additionally, there is a master configuration file that controls settings for the entire TOE. Local parameters must be set on the master and on each segment individually. Master parameters are configured on the master and are either passed to or ignored by the segments at run time.

The TOE defines roles to determine which users can perform operations on parameters, data, and other user roles. Users with a SuperUser role have unlimited access to manage the parameters and data on the TOE and other user accounts. Access to other roles can be given or taken away granularly with the GRANT and REVOKE commands. In this way, each user account can be given a custom set of permissions. Additionally, when a new database object is created, the user account that created the object is given Object Owner permissions to that object by default. Object Owners can control and revoke what commands other users can run against the object.
TOE Security Functional Requirements Satisfied: FMT_MOF.1, FMT_MSA.1, FMT_MSA_(EXT).3, FMT_MTD.1,FMT_REV.1(1), FMT_REV.1(2), FMT_SMF.1, FMT_SMR.1.

7.1.5 Protection of the TSF

The TOE supports mirroring of the master component for high availability. The master keeps a set of metadata that records what data is stored on each segment. This metadata is mirrored to the backup master by a replication process. After failover occurs, the failover process rebuilds the metadata on the backup master to allow operations to resume. Failover is a manual process that must be initiated by an administrator.
TOE Security Functional Requirements Satisfied: FPT_FLS.1, FPT_TRC_(EXT).1.

7.1.6 Resource Utilization

The TOE runs on a distributed architecture, so if the master fails then the segments and back-end network continue to operate (although no processing can take place until the master is restored). Once the master failover takes place, the TOE can immediately resume operations.

The TOE enforces resource queues on users running jobs. Each user is assigned a resource queue when the user's account is created via the "RESOURCE QUEUE" parameter. If a user is not explicitly assigned a queue, then that user's account is assigned to the default "pg_default" queue.

Each resource queue can be configured with a "MAX_COST", "ACTIVE_STATEMENTS", and "MEMORY_LIMIT" parameter. The "MAX_COST" parameter determines the maximum value, in disk page fetches, available for the queue. The parameter can be further configured to OVERCOMMIT, which allows users to wait until the max cost of their queue is low enough for their query to run. If OVERCOMMIT is not set the query is rejected and an error is recorded. The "ACTIVE_STATEMENTS" parameter determines the maximum number of jobs that can be running at one time from the queue. The "MEMORY_LIMIT" parameter determines the maximum amount of memory that all queries submitted through a resource queue can consume on a segment host. Once a queue meets or exceeds these parameters, any additional jobs on that queue are forced to wait until one of the active jobs is completed.
TOE Security Functional Requirements Satisfied: FRU_FLT.1, FRU_RSA.1.

7.1.7 TOE Access

The TOE has the capability to limit concurrent sessions for users and administrators connecting to the master. By default no limit is set for connections, but any SuperUser administrator or administrators with sufficient privileges are able to add a concurrent session limit on a per-user basis. This limit is set via the CONNECTION LIMIT parameter associated with the account ID.

The TOE keeps a log of the date and time of each administrator's and user's previous successful login. This login date and time is presented to the administrator upon request. The administrator can also request the date and time of the last unsuccessful login.

When setting up a user account, administrators typically define a password for the account (if password authentication is to be used). If the password is left blank, it receives a null value. Users and

administrators cannot log in to an account with a null password. Typically this is used to define roles for groups, since group roles are not intended to be logged into with individual user accounts.

Client access and authentication is controlled by a global configuration file on the master node. Access to the TOE is denied unless the role and authentication-method match what is stored in this file. An authorized administrator can modify this file to deny session establishment based on time, user ID, or group membership. Groups are created in the TOE by creating a role and adding multiple users to the role. There is a role ID that can be used to deny session establishment to all users assigned to that role.
TOE Security Functional Requirements Satisfied: FTA_MCS.1, FTA_TAH_(EXT).1, FTA_TSE.1.

8 Rationale

8.1 Conformance Claims Rationale

This Security Target conforms to Part 2 and Part 3 of the Common Criteria Standard for Information Technology Security Evaluations, version 3.1 revision 3.

8.1.1 Protection Profile Conformance

This Security Target conforms to the DBMS PP, with the exception of the FDP_RIP.1 claim. The TOE operating system zeroizes all memory resources used upon completion or termination of a job. Although the TOE itself does not perform this task, the TOE's operational environment performs it on behalf of the TOE. This satisfies the DBMS PP claim of FDP_RIP.1.

8.2 Security Objectives Rationale

This section provides a rationale for the existence of each threat, policy statement, and assumption that compose the Security Target. Sections 8.2.1, 8.2.2, and 8.2.3 demonstrate the mappings between the threats, policies, and assumptions to the security objectives are complete. The following discussion provides detailed evidence of coverage for each threat, policy, and assumption.

8.2.1 Security Objectives Rationale Relating to Threats

Table 17 below provides a mapping of the objects to the threats they counter.

Table 17 – Threats:Objectives Mapping

Threats	Objectives	Rationale
T.ACCOUNTABILITY An administrator might not be able to determine the user responsible for malicious actions that degrade TOE functions.	O.AUDIT_GENERATION The TOE will provide the capability to detect and create records of security relevant events associated with users.	O.AUDIT_GENERATION counters this threat by providing the authorized security administrator with the capability of configuring the audit mechanism to record the actions of a specific user, or review the audit trail based on the identity of the user. Additionally, the security administrator's ID is recorded when any security relevant change is made to the TOE.
	O.TOE_ACCESS The TOE will provide mechanisms that control a user's logical access to the TOE.	O.TOE_ACCESS counters this threat by requiring the TOE to identify and authenticate all authorized users prior to allowing any TOE access or any TOE mediated access on behalf of those users.
	OE.TIME_STAMPS The operational environment will	OE.TIME_STAMPS counters this threat by requiring the IT

Threats	Objectives	Rationale
	provide reliable time stamps.	Environment to provide a reliable time stamp. The audit mechanism is required to include the current date and time in each audit record. All audit records that include the user ID, will also include the date and time that the event occurred.
T.AUDIT_COMPROMISE A user or process may view audit records, cause audit records to be lost or modified, or prevent future audit records from being recorded, thus masking a user's action.	O.AUDIT_REVIEW The TOE will contain mechanisms to allow the authorized security administrator to view and sort the audit logs.	O.AUDIT_REVIEW counters this threat by ensuring that the TOE will provide mechanisms to review the audit logs. These requirements will ensure the data is in a suitable manner for the security administrator to interpret as well as giving the security administrator a way to search and sort within the log to find appropriate data.
	O.AUDIT_STORAGE The TOE will contain mechanisms to provide secure storage and management of the audit log.	O.AUDIT_STORAGE counters this threat by ensuring that the TOE will provide a secure mechanism for storing and managing the TOE audit log.
	O.MANAGE The TOE will provide all the functions and facilities necessary to support the authorized security administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.	O.MANAGE counters this threat by ensuring that the TOE will provide all the functions and facilities necessary to support the authorized security administrator in the management of the security of the audit logs, and restrict these functions and facilities from unauthorized use.
T.MASQUERADE A user or process may masquerade as another entity in order to gain unauthorized access to data or TOE resources.	O.I&A The TOE will contain identification and authentication mechanisms for users to login to the TOE.	O.I&A counters this threat by ensuring that the TOE will contain identification and authentication mechanisms for users to login to the TOE.
	O.TOE_ACCESS The TOE will provide mechanisms that control a user's logical access to the TOE.	O.TOE_ACCESS mitigates this threat by controlling the logical access to the TOE and its resources. By constraining how and when authorized users can access the TOE, and by mandating the type and strength of the authentication mechanisms this objective helps mitigate the possibility of a user attempting to login and masquerade as an authorized user. In addition, this

Threats	Objectives	Rationale
		objective provides the administrator the means to control the number of failed login attempts a user can generate before an account is locked out, further reducing the possibility of a user gaining unauthorized access to the TOE.
T.UNAVAILABILITY The TOE may be overwhelmed by legitimate user tasks, preventing or delaying any TOE functionality from being accessed.	O.QUOTAS The TOE will provide the ability to define quotas for usage of TOE resources, and prevent further allocation of resources once the quotas are reached.	O.QUOTAS counters this threat by ensuring that the TOE limits the amount of jobs that can be operating at one time.
T.TSF_COMPROMISE A malicious user or process may cause configuration data to be inappropriately accessed (viewed, modified or deleted).	O.INTERNAL_TOE_DOMAINS The TSF will maintain internal domains for separation of data and queries belonging to concurrent users.	O.INTERNAL_TOE_DOMAINS ensures that the TOE will establish separate domains for data belonging to users.
	O.MANAGE The TOE will provide all the functions and facilities necessary to support the authorized security administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.	O.MANAGE counters this threat by providing an access control policy that is specified to control access to TSF data. This objective is used to dictate who is able to view and modify TSF data, as well as the behavior of TSF functions.
	O.PARTIAL_SELF_PROTECTIO N 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.PARTIAL_SELF_PROTECTIO N ensures that the TOE is capable of protecting itself from attack.
	O.RESIDUAL_INFORMATION The TOE will ensure that any information contained within its Scope of Control is not released when the resource is reallocated.	O.RESIDUAL_INFORMATION counters this threat by providing security mechanisms that do not allow a user to explicitly view TSF data. If TSF data were to reside inappropriately in a resource that was made available to a user, that user would be able to view the TSF data without authorization.
T.UNAUTHORIZED_ACCESS A user may gain unauthorized access to user data for which they	O.ACCESS_HISTORY The TOE will store and retrieve information (to authorized users)	O.ACCESS_HISTORY is important to mitigate this threat because it ensures the TOE will

Threats	Objectives	Rationale
are not authorized according to the TOE security policy.	related to previous attempts to establish a session.	be able to store and retrieve the information that will advise the user of the last successful login attempt and performed actions without their knowledge.
	O.MEDIATE The TOE must protect user data in accordance with its security policy.	O.MEDIATE counters this threat by ensuring that all accesses to user data are subject to mediation, unless said data has been specifically identifies as public data. The TOE requires successful authentication to the TOE prior to gaining access to any controlled-access content. Lastly, the TSF will ensure that all configured enforcement functions (authentication, access control rules, etc.) must be invoked prior to allowing a user to gain access to TOE or TOE mediated services. The TOE restricts the ability to modify the security attributes associated with access control rules, access to authenticated and unauthenticated services, etc. to the security administrator. This feature ensures that no other user can modify the information flow policy to bypass the intended TOE security policy.
T.UNIDENTIFIED_ACTIONS Failure of the authorized security administrator to identify and act upon unauthorized actions may occur.	O.ADMIN_GUIDANCE The TOE will provide administrators with the necessary information for secure management.	The threat of an authorized administrator failing to know about malicious audit events produces the objectives of the authorized administrator having the facilities and knowing how to use them (O.ADMIN_GUIDANCE).
	O.MANAGE The TOE will provide all the functions and facilities necessary to support the authorized security administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.	O.MANAGE counters this threat by ensuring that authorized administrators have the capability to use the TOE to review audit records.
T.CRITICAL_FAILURE The TOE may experience a failure	O.FAIL_SECURE The TOE will provide mechanisms	O.FAIL_SECURE counters this threat by ensuring that the TOE

Threats	Objectives	Rationale
of a critical component that prevents users and administrators from being able to access TOE functionality.	to allow for secure failure and recovery.	can recover securely from a critical failure.
T.ACCIDENTAL_ADMIN_ERROR An administrator may incorrectly install or configure the TOE resulting in ineffective security mechanisms.	O.ADMIN_GUIDANCE The TOE will provide administrators with the necessary information for secure management.	O.ADMIN_GUIDANCE helps to mitigate this threat by ensuring the TOE administrators have guidance that instructs them how to administer the TOE in a secure manner. Having this guidance helps to reduce the mistakes that an administrator might make that could cause the TOE to be configured insecurely.
T.POOR_DESIGN Unintentional errors in requirements specification or design of the TOE may occur, leading to flaws that may be exploited by a casually mischievous user or program.	O.DOCUMENTED_DESIGN The design of the TOE is adequately and accurately documented.	O.DOCUMENTED_DESIGN ensures that the design of the TOE is documented, permitting detailed review by evaluators.
	O.VULNERABILITY_ANALYSIS The TOE will undergo some vulnerability analysis to demonstrate that the design and implementation of the TOE does not contain any obvious flaws.	O.VULNERABILITY_ANALYSIS ensures that the design of the TOE is analyzed for design flaws.
T.POOR_IMPLEMENTATION Unintentional errors in implementation of the TOE design may occur, leading to flaws that may be exploited by a casually mischievous user or program.	O.CONFIGURATION_IDENTIFICATION The configuration of the TOE is fully identified in a manner that will allow implementation errors to be identified and corrected with the TOE being redistributed promptly.	O.CONFIGURATION_IDENTIFICATION plays a role in countering this threat by requiring the developer to provide control of the changes made to the TOE's design, although the previous three objectives help minimize the introduction of errors into the implementation.
	O.PARTIAL_FUNCTIONAL_TEST The TOE will undergo some security functional testing that demonstrates that the TSF satisfies some of its security functional requirements.	O.PARTIAL_FUNCTIONAL_TEST increases the likelihood that any errors that do exist in the implementation (with respect to the functional specification, high-level, and low-level design) will be discovered through testing.
	O.VULNERABILITY_ANALYSIS The TOE will undergo some vulnerability analysis to demonstrate that the design and implementation of the TOE does not contain any obvious flaws.	O.VULNERABILITY_ANALYSIS helps reduce errors in the implementation that may not be discovered during functional testing. Ambiguous design documentation and the fact that exhaustive testing of the external interfaces is not required may leave bugs in the implementation

Threats	Objectives	Rationale
		undiscovered in functional testing.
T.POOR_TEST Lack of or insufficient tests to demonstrate that all TOE security functions operate correctly (including in a fielded TOE) may result in incorrect TOE behavior being discovered thereby causing potential security vulnerabilities.	O.DOCUMENTED_DESIGN The design of the TOE is adequately and accurately documented.	O.DOCUMENTED_DESIGN helps to ensure that the TOE's documented design satisfies the security functional requirements. In order to ensure the TOE's design is correctly realized in its implementation, the appropriate level of functional testing of the TOE's security mechanisms must be performed during the evaluation of the TOE.
	O.PARTIAL_FUNCTIONAL_TEST The TOE will undergo some security functional testing that demonstrates that the TSF satisfies some of its security functional requirements.	O.PARTIAL_FUNCTIONAL_TEST increases the likelihood that any errors that do exist in the implementation (with respect to the functional specification, high-level, and low-level design) will be discovered through testing.
	O.VULNERABILITY_ANALYSIS The TOE will undergo some vulnerability analysis to demonstrate that the design and implementation of the TOE does not contain any obvious flaws.	O.VULNERABILITY_ANALYSIS addresses this concern by requiring a vulnerability analysis be performed in conjunction with testing that goes beyond functional testing. This objective provides a measure of confidence that the TOE does not contain security flaws that may not be identified through functional testing. While these testing activities are a necessary activity for successful completion of an evaluation, this testing activity does not address the concern that the TOE continues to operate correctly and enforce its security policies once it has been fielded. Some level of testing must be available to end users to ensure the TOE's security mechanisms continue to operate correctly once the TOE is fielded.
T.RESIDUAL_DATA A user or process may gain unauthorized access to data through reallocation of TOE resources from one user or process to another.	O.RESIDUAL_INFORMATION The TOE will ensure that any information contained within its Scope of Control is not released when the resource is reallocated.	O.RESIDUAL_INFORMATION counters this threat by ensuring that TSF data and user data is not persistent when resources are released by one user/process and allocated to another user/process.

Every Threat is mapped to one or more Objectives in the table above. This complete mapping demonstrates that the defined security objectives counter all defined threats.

8.2.2 Security Objectives Rationale Relating to Policies

Table 18 below gives a mapping of policies and the objectives that support them.

Table 18 - Policies:Objectives Mapping

Policies	Objectives	Rationale
P.ACCOUNTABILITY The authorized users of the TOE shall be held accountable for their actions within the TOE.	O.AUDIT_GENERATION The TOE will provide the capability to detect and create records of security relevant events associated with users.	O.AUDIT_GENERATION addresses this policy by providing the authorized administrator with the capability of configuring the audit mechanism to record the actions of a specific user, or review the audit trail based on the identity of the user. Additionally, the administrator's ID is recorded when any security relevant change is made to the TOE (e.g., access rule modification, start-stop of the audit mechanism, establishment of a trusted channel, etc.).
	O.TOE_ACCESS The TOE will provide mechanisms that control a user's logical access to the TOE.	O.TOE_ACCESS supports this policy by requiring the TOE to identify and authenticate all authorized users prior to allowing any TOE access or any TOE mediated access on behalf of those users.
P.ROLES The TOE shall provide an authorized administrator role for secure administration of the TOE. This role shall be separate and distinct from other authorized users.	O.ADMIN_ROLE The TOE will provide authorized administrator roles to isolated administrative actions.	The TOE has the objective of providing an authorized administrator role for secure administration. The TOE may provide other roles as well, but only the role of authorized administrator is required (O.ADMIN_ROLE).

Every policy is mapped to one or more Objectives in the table above. This complete mapping demonstrates that the defined security objectives enforce all defined policies.

8.2.3 Security Objectives Rationale Relating to Assumptions

Table 19 below gives a mapping of assumptions and the environmental objectives that uphold them.

Table 19 – Assumptions:Objectives Mapping

Assumptions	Objectives	Rationale
A.DOMAIN_SEPARATION The operational environment will provide a separate domain for the TOE's operation.	OE.DOMAIN_SEPARATION The operational environment will provide an isolated domain for the execution of the TOE.	OE.DOMAIN_SEPARATION upholds this assumption by ensuring that the operational environment will provide an isolated domain for the TOE's execution.
A.I&A The operational environment will provide identification and authentication mechanisms for use of utilities under the control of the operational environment.	OE.I&A The operational environment will contain identification and authentication mechanisms for administrator access to database control utilities and other utilities.	OE.I&A upholds this assumption by ensuring that the operational environment will provide mechanisms for administrators to be authenticated before any database control utilities and other utilities used to manage system resources and I/O interfaces may be used.
A.NO_BYPASS The operational environment will ensure the TSF cannot be bypassed in order to gain access to TOE data.	OE.NO_BYPASS The operational environment shall ensure the TOE security mechanisms cannot be bypassed in order to gain access to the TOE resources.	OE.NO_BYPASS upholds this assumption by ensuring that the TOE cannot be bypassed in order to gain unauthorized access of TOE resources.
A.NO_EVIL Administrators are non-hostile, appropriately trained, and follow all administrator guidance.	NOE.NO_EVIL Sites using the TOE shall ensure that authorized administrators are non-hostile, appropriately trained, and follow all administrator guidance.	All authorized administrators are trustworthy individuals, having background investigators commensurate with the level of data being protected, have undergone appropriate admin training, and follow all admin guidance.
A.NO_GENERAL_PURPOSE There are no general-purpose computing capabilities (e.g., compilers or user applications) available on the DBMS servers, other than those services necessary for the operation, administration, and support of the DBMS.	OE.NO_GENERAL_PURPOSE There will be no general-purpose computing capabilities (e.g., compilers or user applications) available on DBMS servers, other than those services necessary for the operation, administration and support of the DBMS.	The DBMS server must not include any general-purpose commuting or storage capabilities. This will protect the TSF data from malicious processes.
A.PHYSICAL It is assumed that appropriate physical security is provided within the domain for the value of the IT assets protected by the TOE and the value of the stored, processed, and transmitted information.	NOE.PHYSICAL Physical security will be provided within the domain for the value of the IT assets protected by the TOE and the value of the stored, processed, and transmitted information.	The TOE, the TSF data, and protected user data is assumed to be protected from physical attack (e.g., theft, modification, destruction, or eavesdropping). Physical attack could include unauthorized intruders into the TOE environment, but it does not include physical destructive actions that might be taken by an

Assumptions	Objectives	Rationale
		individual that is authorized to access the TOE environment.
A.RESTRICT_OS_ACCESS Logon access to the underlying operating system is restricted to authorized administrators only.	OE.RESTRICT_OS_ACCESS The underlying operating system will be configured with only those user accounts required for access by authorized security administrators.	OE.RESTRICT_OS_ACCESS upholds this assumption by ensuring that the underlying operating system running on the RDBMS server must include only those user accounts required by authorized administrators. Restricting access to the operating system protects against tampering by malicious users.
A.ROBUST_ENVIRONMENT The operational environment is at least as robust as the TOE.	OE.ROBUST_ENVIRONMENT The operational environment that supports the TOE for enforcement of its security objectives will be of at least the same level of robustness as the TOE.	OE.ROBUST_ENVIRONMENT upholds this assumption by ensuring that the TOE shall only be installed in an operational environment that is at least as robust as the TOE. The TOE is basic-enhanced robustness, therefore, all elements in the environment the TOE depends on for enforcement of its security objectives are also assumed to be basic-enhanced robustness. These elements could include the operating system, encryption devices, and/or boundary protection devices.
	OE.TRUST_IT Each operational entity the TOE relies on for security functions will be installed, configured, managed and maintained in a manner appropriate to the IT entity, and consistent with the security policy of the TOE and the relationship between them.	OE.TRUST_IT upholds this assumption by ensuring that the IT entities in the environment are correctly installed, configured, managed and maintained.
A.SECURE_COMMS The operational environment will provide a secure (protected from disclosure, spoofing, and able to detect modification) line of communications between the remote user and the TOE.	OE.SECURE_COMMS The operational environment will provide a secure line of communications between the remote user and the TOE.	OE.SECURE_COMMS upholds this assumption by stating that the environment must provide a secure line of communication for transfer of TSF data. This is necessary because access to the TOE may be distributed geographically with users and authorized administrators in different locations. The objective OE.SECURE_COMMS does not necessarily mandate that the communications between the

Assumptions	Objectives	Rationale
		remote user or administrator and the TOE be encrypted.
A.TIME_STAMPS The operational environment will provide the TOE with the necessary reliable timestamps.	OE.TIME_STAMPS The operational environment will provide reliable time stamps.	OE.TIME_STAMPS upholds this assumption by stating that the environment will maintain reliable timestamps and those will be used by the TOE to stamp each audit record with a date and time.

Every assumption is mapped to one or more Objectives in the table above. This complete mapping demonstrates that the defined security objectives uphold all defined assumptions.

8.3 Rationale for Extended Security Functional Requirements

Table 17 presents the rationale for the inclusion of the extended functional and assurance requirements found in the DBMS PP. The extended requirements that are included as NIAP interpretations do not require a rationale for their inclusion per CCEVS management.

Table 20 – Rationale for Extended Requirements

Extended Requirement	Identifier	Rationale
FAU_GEN_(EXT).2	User and/or group identity association	This requirement was needed to replace FAU_GEN.2.1-NIAP-0410 because the DBMS PP does not require the TOE to implement a user identity and/or a group identity to satisfy the DAC policy. Therefore, this extended requirement was created to allow the audit function to use the user identity or the group identity or both.
FPT_TRC_(EXT).1	Internal TSF consistency	FPT_TRC_(EXT).1 has been created to require timely consistency of replicated TSF data. Although there is a Common Criteria Requirement that attempts to address this functionality, it falls short of the needs of the environment in the DBMS PP. Specifically, FPT_TRC.1.1 states "The TSF shall ensure that TSF data is consistent when replicated between parts of the TOE." In the widely distributed environment of the DBMS PP's TOE, this is an infeasible requirement. For TOEs with a very large number of components, 100 percent TSF data consistency is not achievable and is not expected at any specific instant in time. Another concern lies in FPT_TRC.1.2 that states that when replicated parts of the TSF are "disconnected", the TSF shall ensure consistency of the TSF replicated data upon "reconnection". Upon first inspection, this

Extended Requirement	Identifier	Rationale
		seems reasonable, however, when applying this requirement it becomes clear that it dictates specific mechanisms to determine when a component is "disconnected" from the rest of the TSF and when it is "reconnected". This is problematic in the DBMS PP's environment in that it is not the intent of the authors to dictate that distributed TSF components keep track of connected/disconnected components. In general, to meet the needs of this PP, it is acceptable to only require a mechanism that provides TSF data consistency in a timely manner after it is determined that it is inconsistent.
FTA_TAH_(EXT).1	TOE Access History	The DBMS PP does not require the TOE to contain a client. Therefore, the PP cannot require the client to dispay a message. This requirement has been modified to require the TOE to store and retrieve the access history instead of displaying it.
FMT_MSA_(EXT).3	Static attribute initialization	The CC does not allow the PP author to specify restrictive values that are not modifiable. This extended requirement eliminates the element FMT_MSA.3.2 from the component FMT_MSA.3 and makes the component more secure by requiring the security attributes of the objects on creation to be restrictive and not allowing any user to be able to override the restrictive default values.

8.4 Rationale for Extended TOE Security Assurance Requirements

No extended SARs have been defined for this ST.

8.5 Security Requirements Rationale

The following discussion provides detailed evidence of coverage for each security objective.

8.5.1 Rationale for Security Functional Requirements of the TOE Objectives

Table 21 below shows a mapping of the objectives and the SFRs that support them.

Table 21 – Objectives:SFRs Mapping

Objective	Requirements Addressing the Objective	Rationale
O.AUDIT_GENERATION The TOE will provide the capability to detect and create records of security relevant events associated with users.	FAU_GEN.1-NIAP-0410 Audit Data Generation	FAU_GEN.1-NIAP-0410 defines the set of events that the TOE must be capable of recording. This requirement ensures that the administrator has the ability to audit any security-relevant events that take place on the TOE. This requirement also defines the information that must be contained in the audit record for each auditable event. This requirement also places a requirement on the level of detail that is recorded on any additional security functional requirements an ST author adds to this PP.
	FAU_GEN_(EXT).2 User and/or group identity association	FAU_GEN_(EXT).2 ensures that the audit records associate a user and/or group identity with the auditable event. In the case of authorized users, the association is accomplished with the user ID. In the case of authorized groups, the association is accomplished with the group ID.
O.AUDIT_REVIEW The TOE will contain mechanisms to allow the authorized security administrator to view and sort the audit logs.	FAU_SAR.1 Audit review	FAU_SAR.1 supports this objective by requiring that only the authorized security administrator has the capability to read the audit records which must be presented in a manner suitable for the security administrator to interpret them.
	FAU_SAR.2 Restricted audit review	FAU_SAR.2 supports this objective by prohibiting all other users read access of the audit records.
	FAU_SAR.3 Selectable audit review	FAU_SAR.3 supports this objective by requiring the TOE to provide a mechanism for the security administrator to search and sort through the audit records.
O.AUDIT_GENERATION The TOE will provide the capability to detect and create records of security relevant events associated	FAU_SEL.1-NIAP-0407 Selective audit	FAU_SEL-NIAP-0407 allows the administrator to configure which auditable events will be recorded in the audit trail. This provides

Objective	Requirements Addressing the Objective	Rationale
with users.		the administrator with the flexibility in recording only those events that are deemed necessary by site policy, thus reducing the amount of resources consumed by the audit mechanism.
O.AUDIT_STORAGE The TOE will contain mechanisms to provide secure storage and management of the audit log.	FAU_STG.1 Protected audit trail storage	FAU_STG.1 supports this objective by requiring that only the authorized security administrator may delete the audit records ensuring that no malicious users may compromise the data stored within the audit records.
O.MEDIATE The TOE must protect user data in accordance with its security policy.	FDP_ACC.1 Subset access control	The FDP requirements were chosen to define the policies, the subjects, objects, and operations for how and when mediation takes place in the TOE. FDP_ACC.1 defines the Access Control policy that will be enforced on a list of subjects acting on the behalf of users attempting to gain access to a list of named objects. All the operations between subjects and objects covered are defined by the TOE's policy.
	FDP_ACF.1-NIAP-0407 Security attribute based access control	FDP_ACF.1-NIAP-0407 defines the security attributes used to provide access control to objects based on the TOE's access control policy.
O.RESIDUAL_INFORMATION The TOE will ensure that any information contained within its Scope of Control is not released when the resource is reallocated.	FDP_RIP.1 Subset residual information protection	FDP_RIP.1 is used to ensure the contents of processes are not available to subjects other than those explicitly granted access to the data. Each session in the TOE uses unique processes and the TOE environment allocates memory to each unique process.
O.TOE_ACCESS The TOE will provide mechanisms that control a user's logical access to the TOE.	FIA_ATD.1 User attribute definition	FIA_ATD.1 defines the attributes of users, including a user ID that is used by the TOE to determine a user's identity and/or group memberships and enforce what type of access the user has to the TOE.

Objective	Requirements Addressing the Objective	Rationale
O.I&A The TOE will contain identification and authentication mechanisms for users to login to the TOE.	FIA_ATD.1 User attribute definition	FIA_ATD.1 supports this objective by requiring the TOE to maintain user identities and passwords belonging to individual users.
	FIA_UAU.2 User authentication before any action	FIA_UAU.1 supports this objective by requiring the TOE to successfully authenticate a user before establishing a session on behalf of the user.
	FIA_UID.2 User identification before any action	FIA_UID.1 supports this objective by requiring the TOE to successfully identify a user before establishing a session on behalf of the user.
O.TOE_ACCESS The TOE will provide mechanisms that control a user's logical access to the TOE.	FIA_USB.1 User-subject binding	FIA_USB.1 supports this objective by ensuring that all subjects that act on behalf of users will have a binding that associates the subjects with a user uniquely.
O.I&A The TOE will contain identification and authentication mechanisms for users to login to the TOE.	FIA_USB.1 User-subject binding	FIA_USB.1 supports this objective by requiring that all subjects that act on behalf of users must have a binding that associates the subjects with a user uniquely.
O.AUDIT_GENERATION The TOE will provide the capability to detect and create records of security relevant events associated with users.	FIA_USB.1 User-subject binding	FIA_USB.1 supports this objective by requiring that all subjects that act on behalf of users must have a binding that associates the subjects with a user. This is necessary to be able to associate audit records with user identities.
O.MANAGE The TOE will provide all the functions and facilities necessary to support the authorized security administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.	FMT_MOF.1 Management of security functions behaviour	FMT_MOF.1 requires that the ability to use particular TOE capabilities be restricted to the administrator.
	FMT_MSA.1 Management of security attributes	FMT_MSA.1 requires that the ability to perform operations on security attributes be restricted to particular roles.
	FMT_MSA_(EXT).3 Static attribute initialisation	FMT_MSA_(EXT).3 requires that default values used for security attributes are restrictive.
	FMT_MTD.1 Management of TSF data	FMT_MTD.1 requires that the ability to manipulate TOE content is restricted to administrators.

Objective	Requirements Addressing the Objective	Rationale
	FMT_REV.1(1) Revocation	FMT_REV.1 restricts the ability to revoke attributes to the administrator.
	FMT_REV.1(2) Revocation	FMT_REV.1 restricts the ability to revoke attributes to the administrator.
O.AUDIT_STORAGE The TOE will contain mechanisms to provide secure storage and management of the audit log.	FMT_SMF.1 Specification of management functions	FMT_SMF.1 supports this objective by listing the mechanisms available to the security administrator for managing the audit records.
O.MANAGE The TOE will provide all the functions and facilities necessary to support the authorized security administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.	FMT_SMF.1 Specification of management functions	FMT_SMF.1 identifies the management functions that are available to the authorized administrator.
	FMT_SMR.1 Security roles	FMT_SMR.1 defines the specific security roles to be supported.
O.ADMIN_ROLE The TOE will provide authorized administrator roles to isolated administrative actions.	FMT_SMR.1 Security roles	The TOE will establish, at least, an authorized administrator role. The ST writer may choose to specify more roles. The authorized administrator will be given privileges to perform certain tasks that other users will not be able to perform. These privileges include, but are not limited to, access to audit information and security functions. (FMT_SMR.1)
O.FAIL_SECURE The TOE will provide mechanisms to allow for secure failure and recovery.	FPT_FLS.1 Failure with preservation of secure state	FPT_FLS.1 supports this objective by ensuring that the TOE can enter a secure state after the failure of a master.
O.MEDIATE The TOE must protect user data in accordance with its security policy.	FPT_TRC_(EXT).1 Internal TSF consistency	Replicated TSF data that specifies attributes for access control must be consistent across distributed components of the TOE. The requirement is to maintain consistency of replicated TSF data.
O.FAIL_SECURE The TOE will provide mechanisms to allow for secure failure and recovery.	FRU_FLT.1 Degraded fault tolerance	FRU_FLT.1 supports this objective by ensuring that the TOE can recover from the failure of a master.
O.QUOTAS The TOE will provide the ability to	FRU_RSA.1 Maximum quotas	FRU_RSA.1 supports this objective by allowing the TOE to

Objective	Requirements Addressing the Objective	Rationale
define quotas for usage of TOE resources, and prevent further allocation of resources once the quotas are reached.		define maximum quotas on usage of TOE resources.
O.TOE_ACCESS The TOE will provide mechanisms that control a user's logical access to the TOE.	FTA_MCS.1 Basic limitation on multiple concurrent sessions	FTA_MCS.1 ensures that users may only have a maximum of a specified number of active sessions open at any given time.
O.INTERNAL_TOE_DOMAINS The TSF will maintain internal domains for separation of data and queries belonging to concurrent users.	FTA_MCS.1 Basic limitation on multiple concurrent sessions	FTA_MCS.1 ensures that the TOE can provide multiple concurrent sessions. This along with ADV.ARC.1 ensure that domain separation exists to securely implement concurrent sessions within the TOE.
O.ACCESS_HISTORY The TOE will store and retrieve information (to authorized users) related to previous attempts to establish a session.	FTA_TAH_(EXT).1 TOE access history	The TOE must be able to store and retrieve information about previous unauthorized login attempts and the number of times the login was attempted every time the user logs into their account. The TOE must also store the last successful authorized login. This information will include the date, time, method, and location of the attempts. When appropriately displayed, this will allow the user to detect if another use is attempting to access their account. These records should not be deleted until after the user has been notified of their access history. (FTA_TAH_(EXT).1)
O.TOE_ACCESS The TOE will provide mechanisms that control a user's logical access to the TOE.	FTA_TSE.1 TOE session establishment	FTA_TSE.1 allows the TOE to restrict access to the TOE based on certain criteria.

Every objective is mapped to one or more functional requirements in the table above. This complete mapping demonstrates that the defined functional requirements support all defined Objectives.

8.5.2 Security Assurance Requirements Rationale

EAL2+ was chosen for this ST to conform to the DBMS PP. The DBMS PP was developed for use by Commercial DBMS Security Software developers.

Flaw Remediation is the only requirement not included in any EAL because it does not add any assurance to the current system, but to subsequent releases. Therefore, the PP Review Board decided to augment EAL2 with ALC-FLR.2 to instruct the vendors on proper flaw remediation techniques.

8.5.3 Rationale for Security Assurance Requirements of the TOE Objectives

Table 22 below shows a mapping of the objectives and the SARs that support them.

Table 22 – Objectives:SARs Mapping

Objective	Requirements Addressing the Objective	Rationale
O.INTERNAL_TOE_DOMAINS	ADV_ARC.1	ADV_ARC.1 provides the security architecture description of the security domains maintained by the TSF that are consistent with the SFRs. Since self-protection is a property of the TSF that is achieved through the design of the TOE and TSF, and enforced by the correct implementation of that design, self-protection will be achieved by that design and implementation.
O.PARTIAL_SELF_PROTECTION	ADV_ARC.1	ADV_ARC.1 provides the security architecture description of the security domains maintained by the TSF that are consistent with the SFRs. Since self-protection is a property of the TSF that is achieved through the design of the TOE and TSF, and enforced by the correct implementation of that design, self-protection will be achieved by that design and implementation.
O.DOCUMENTED_DESIGN	ADV_FSP.2	ADV_FSP.2 requires that the interfaces to the TOE be documented and specified.
	ADV_TDS.1-1	ADV_TDS.1 requires the high-level design of the TOE be documented and specified and that said design be shown to correspond to the interfaces.
	ADV_TDS.1-2	ADV_TDS.1 requires that there be a correspondence between adjacent layers of the design decomposition.
O.ADMIN_GUIDANCE	AGD-OPE.1-1	AGD_OPE.1 mandates the developer provide the administrator with guidance on how to operate the TOE in a

Objective	Requirements Addressing the Objective	Rationale
		secure manner. This includes describing the interfaces the administrator uses in managing the TOE, security parameters that are configurable by the administrator, how to configure the TOE's rule set and the implications of any dependencies of individual rules. The documentation also provides a description of how to setup and review the auditing features of the TOE. The guidance must show the administrator how to use the functionality available, review the results of any tests and/or alerts, and act accordingly.
	AGD_OPE.1-2	AGD_OPE.1 is also intended for non-administrative users, but it could be used to provide guidance on security that is common to both administrators and non-administrators (e.g., password management guidelines).
	AGD_OPE.1-3	AGD_OPE.1 and AGD_PRE.1 analysis during the evaluation will ensure that the guidance documentation is complete and consistent, and notes all requirements for external security measures.
	AGD_PRE.1-1	AGD_OPE.1 and AGD_PRE.1 analysis during the evaluation will ensure that the guidance documentation is complete and consistent, and notes all requirements for external security measures.
	AGD_PRE.1-2	AGD_PRE.1 ensures the administrator has the information necessary to install the TOE in the evaluated configuration. Often times a vendor's product contains software that is not part of the TOE and has not been evaluated. The Preparative User Guidance (AGD_PRE) documentation ensures that once the administrator has followed the installation and configuration

Objective	Requirements Addressing the Objective	Rationale
		guidance the result is a TOE in a secure configuration.
	ALC_DEL.1	ALC_DEL.1 ensures that the administrator is provided documentation that instructs them how to ensure the delivery of the TOE, in whole or in parts, has not been tampered with or corrupted during delivery. This requirement ensures the administrator has the ability to begin their TOE installation with a clean (e.g., malicious code has not been inserted once it has left the developer's control) version of the TOE, which is necessary for secure management of the TOE.
O.CONFIGURATION_IDENTIFICATION	ALC_CMS.2	ALC_CMS.2 addresses this objective by requiring that there be a unique reference for the TOE, and that the TOE is labeled with that reference. It also requires that there be a CM system in place, and that the configuration items that comprise the TOE are uniquely identified. This provides a clear identification of the composition of the TOE.
	ALC_FLR.2	ALC_FLR.2 addresses this objective by requiring that there be a mechanism in place for identifying flaws subsequent to fielding, and for distributing those flaws to entities operating the system.
O.PARTIAL_FUNCTIONAL_TEST	ATE_COV.1	ATE_COV.1 requires that there be a correspondence between the tests in the test documentation and the TSF as described in the functional specification.
	ATE_FUN.1	ATE_FUN.1 requires that the developer provide test documentation for the TOE, including test plans, test procedure descriptions, expected test results, and actual test results. These need to identify the functions tested, the tests performed, and test scenarios.

Objective	Requirements Addressing the Objective	Rationale
		These require that the developer run those tests, and show that the expected results were achieved.
	ATE_IND.2	ATE_IND.2 requires that the evaluators test a subset of the TSF to confirm correct operation, on an equivalent set of resources to those used by the developer for testing. These sets should include a subset of the developer-run tests.
O.VULNERABILITY_ANALYSIS	AVA_VAN.2	The AVA_VAN.2 component provides the necessary level of confidence that vulnerabilities do not exist in the TOE that could cause the security policies to be violated. AVA_VAN.2 requires the evaluator to perform a search for potential vulnerabilities in all the TOE deliverables. For those vulnerabilities that are not eliminated by the developer, a rationale must be provided that describes why these vulnerabilities cannot be exploited by a threat agent with a basic attack potential, which is in keeping with the desired assurance level of this TOE. This component provides the confidence that security flaws do not exist in the TOE that could be exploited by a threat agent of basic attack potential to violate the TOE's security policies.

Every objective is mapped to one or more assurance requirements in the table above. This complete mapping demonstrates that the defined assurance requirements support all defined Objectives.

8.5.4 Dependency Rationale

This ST does satisfy all the requirement dependencies of the Common Criteria. Table 23 lists each requirement to which the TOE claims conformance with a dependency and indicates whether the dependent requirement was included. As the table indicates, all dependencies have been met.

Table 23 – Functional Requirements Dependencies

SFR ID	Dependencies	Dependency Met	Rationale
FAU_GEN.1-NIAP-0410	FPT_STM.1	✓	This requirement must be satisfied by the IT environment because the

SFR ID	Dependencies	Dependency Met	Rationale
			DBMS is a software-only TOE.
FAU_GEN_(EXT).2	FAU_GEN.1	✓	Although FAU_UID.1 is not included, FIA_UID.2, which is hierarchical to FIA_UID.1 is included. This satisfies this dependency.
	FIA_UID.1	✓	Although FIA_UID.1 is not included, FIA_UID.2, which is hierarchical to FIA_UID.1 is included. This satisfies this dependency.
FAU_SAR.1	FAU_GEN.1	✓	
FAU_SAR.2	FAU_SAR.1	✓	
FAU_SAR.3	FAU_SAR.1	✓	
FAU_SEL.1-NIAP-0407	FAU_GEN.1-NIAP-0410	✓	
	FMT_MTD.1	✓	
FAU_STG.1	FAU_GEN.1	✓	
FDP_ACC.1	FDP_ACF.1-NIAP-0407	✓	
FDP_ACF.1-NIAP-0407	FMT_MSA.3	✓	The dependency on FMT_MSA.3 is satisfied by FMT_MSA_(EXT).3.
	FDP_ACC.1	✓	
FDP_RIP.1	None	N/A	
FIA_ATD.1	None	N/A	
FIA_UAU.2	FIA_UID.1	✓	Although FIA_UID.1 is not included, FIA_UID.2, which is hierarchical to FIA_UID.1 is included. This satisfies this dependency.
FIA_UID.2	None	N/A	
FIA_USB.1	FIA_ATD.1	✓	
FMT_MOF.1	FMT_SMR.1	✓	
	FMT_SMF.1	✓	
FMT_MSA.1	FMT_SMF.1	✓	
	FDP_ACC.1	✓	
	FMT_SMR.1	✓	

SFR ID	Dependencies	Dependency Met	Rationale
FMT_MSA_(EXT).3	FMT_MSA.1	✓	
	FMT_SMR.1	✓	
FMT_MTD.1	FMT_SMR.1	✓	
	FMT_SMF.1	✓	
FMT_REV.1(1)	FMT_SMR.1	✓	
FMT_REV.1(2)	FMT_SMR.1	✓	
FMT_SMF.1	None	N/A	
FMT_SMR.1	FIA_UID.1	✓	Although FIA_UID.1 is not included, FIA_UID.2, which is hierarchical to FIA_UID.1 is included. This satisfies this dependency.
FPT_FLS.1	None	N/A	
FPT_TRC_(EXT).1	None	N/A	
FRU_FLT.1	FPT_FLS.1	✓	
FRU_RSA.1	None	N/A	
FTA_MCS.1	FIA_UID.1	✓	Although FIA_UID.1 is not included, FIA_UID.2, which is hierarchical to FIA_UID.1 is included. This satisfies this dependency.
FTA_TAH_(EXT).1	None	N/A	
FTA_TSE.1	None	N/A	

9 Acronyms and Terms

This section and Table 24 define the acronyms and terms used throughout this document.

9.1 Acronyms

Table 24 – Acronyms

Acronym	Definition
BASH	Bourne Again Shell
CC	Common Criteria
CEM	Common Evaluation Methodology
CLI	Command Line Interface
CPU	Central Processing Unit
CSV	Comma Separated Values
DB	Database
EAL	Evaluation Assurance Level
GB	Gigabyte
GCC	GNU Compiler Collection
GNU	GNU's Not Unix
GUI	Graphical User Interface
ID	Identifier
IP	Internet Protocol
IT	Information Technology
JDBC	Java Database Connectivity
LDAP	Lightweight Directory Access Protocol
MB	Megabyte
ODBC	Open DataBase Connectivity
OS	Operating System
PP	Protection Profile
RAM	Random Access Memory
RHEL	Red Hat Enterprise Linux
RDBMS	Relational Database Management System
SAR	Security Assurance Requirement
SFP	Security Functional Policy
SFR	Security Functional Requirement

Acronym	Definition
SLES	SuSE Linux Enterprise Server
SQL	Structured Query Language
ST	Security Target
TOE	Target of Evaluation
TSF	TOE Security Functionality

9.2 Terms

Access – Interaction between an entity and an object that results in the flow or modification of data.

Access Control – Security service that controls the use of resources[14] and the disclosure and modification of data.[15]

Accountability – Property that allows activities in an IT system to be traced to the entity responsible for the activity.

Administrator – A user who has been specifically granted the authority to manage some portion or all of the TOE and whose actions may affect the TSP. Administrators may possess special privileges that provide capabilities to override portions of the TSP.

Assurance – A measure of confidence that the security features of an IT system are sufficient to enforce its" security policy.

Attack – An intentional act attempting to violate the security policy of an IT system.

Authentication – Security measure that verifies a claimed identity.

Authentication data – Information used to verify a claimed identity.

Authorization – Permission, granted by an entity authorized to do so, to perform functions and access data.

Authorized Administrator – The authorized person in contact with the Target of Evaluation who is responsible for maintaining its operational capability.

Authorized user – An authenticated user who may, in accordance with the TSP, perform an operation.

Availability – Timely,[16] reliable access to IT resources.

Compromise – Violation of a security policy.

Confidentiality – A security policy pertaining to disclosure of data.

[14] Hardware and software
[15] Stored or communicated
[16] According to a defined metric

Conformant Product – A Target of Evaluation that satisfied all the functional security requirements in Section 5.1 and satisfies all the TOE security assurance requirements in section 5.2 of the DBMS PP.

Critical Security Parameters (CSP) – Security-related information (e.g., cryptographic keys, authentication data such as passwords and pins, and cryptographic seeds) appearing in plaintext or otherwise unprotected form and whose disclosure or modification can compromise the security of a cryptographic module or the security of the information protected by the module.

Database Management System (DBMS) – A suite of programs that typically manage large structured sets of persistent data, offering ad hoc query facilities to many users. They are widely used in business applications.

Defense-in-Depth (DID) – A security design strategy whereby layers of protection are utilized to establish an adequate security posture for an IT system.

Discretionary Access Control (DAC) – A means of restricting access to objects based on the identity of subjects and/or groups to which they belong. Those controls are discretionary in the sense that a subject with a certain access permission is capable of passing that permission (perhaps indirectly) on to any other subject.

Enclave – A collection of entities under the control of a single authority and having a homogeneous security policy. They may be logical, or may be based on physical location and proximity.

Entity – A subject, object, user or another IT device, which interacts with TOE objects, data, or resources.

External IT entity – Any trusted Information Technology (IT) product or system, outside of the TOE, which may, in accordance with the TSP, perform an operation.

Identity – A representation (e.g., a string) uniquely identifying an authorized user, which can either be the full or abbreviated name of that user or a pseudonym.

Integrity – A security policy pertaining to the corruption of data and TSF mechanisms.

Named Object – An object that exhibits all of the following characteristics: The object may be used to transfer information between subjects of differing user and/or group identities within the TSF. Subjects in the TOE must be able to require a specific instance of the object. The name used to refer to a specific instance of the object must exist in a context that potentially allows subjects with different user and/or group identities to require the same instance of the object.

Object – An entity within the TSC that contains or receives information and upon which subjects perform operations.

Operating Environment – The total environment in which a TOE operates. It includes the physical facility and any physical, procedural, administrative and personnel controls.

Public Object – An object for which the TSF unconditionally permits all entities "read" access. Only the TSF or authorized administrators may create, delete, or modify the public objects.

Secure State – Condition in which all TOE security policies are enforced.

Security attributes – TSF data associated with subjects, objects, and users that are used for the enforcement of the TSP.

Security level – The combination of a hierarchical classification and a set of non-hierarchical categories that represent the sensitivity of the information.

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

Subject – An entity within the TSC that causes operation to be performed.

Threat – Capabilities, intentions and attack methods of adversaries, or any circumstance or event, with the potential to violate the TOE security policy.

Threat Agent – Any human user or Information Technology (IT) product or system, which may attempt to violate the TSP and perform an unauthorized operation with the TOE.

Unauthorized user – A user who may obtain access only to system provided public objects if any exist.

User – Any entity (human user or external IT entity) outside the TOE that interacts with the TOE.

Vulnerability – A weakness that can be exploited to violate the TOE security policy.

Prepared by:
Corsec Security, Inc.

13135 Lee Jackson Memorial Highway, Suite 220
Fairfax, VA 22033
United States of America

Phone: +1 703 267 6050
Email: info@corsec.com
http://www.corsec.com