Netezza Corporation
Netezza Performance Server v4.6.5

Security Target

Evaluation Assurance Level: EAL4+
Document Version: 0.6

Prepared for:

NETEZZA The Power to Question Everything™

Netezza Corporation
26 Forest Street
Marlborough, MA 01752
Phone: (508) 382-8200
Fax: (508)382-8300
http://www.netezza.com

Prepared by:

Corsec SM

Corsec Security, Inc.
10340 Democracy Lane, Suite 201
Fairfax, VA 22030
Phone: (703) 267-6050

http://www.corsec.com

Table of Contents

Table of Figures

Table of Tables

1 Security Target Introduction

This section identifies the Security Target (ST), Target of Evaluation (TOE), and the ST organization. The Target of Evaluation is the Netezza Performance Server v4.6.5 (NPS), and will hereafter be referred to as the TOE throughout this document. The TOE is a data warehousing product that provides support for a wide range of business intelligence applications.

1.1 Purpose

This ST provides mapping of the Security Environment to the Security Requirements that the TOE meets in order to remove, diminish or mitigate the defined threats in the following sections:

- Security Target 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.
- Conformance Claims (Section 2) – Provides the identification of any Common Criteria (CC), ST Protection Profile, and Evaluation Assurance Level (EAL) package claims.
- Security Problem Definition (Section 3) – Describes the threats, 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 Definition (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 Security Functional Requirements (SFRs) and Security Assurance Requirements (SARs) met by the TOE and by the TOE's environment.
- 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 and Terminology (Section 9) – Defines the acronyms and terminology used within this ST.

1.2 Security Target and TOE References

Table 1 - ST and TOE References

ST Title	Netezza Corporation Netezza Performance Server v4.6.5 Security Target
ST Version	Version 0.6
ST Author	Corsec Security, Inc. Amy Nicewick
ST Publication Date	2010-04-13
TOE Reference	Netezza Performance Server v4.6.5.build 10670
Keywords	Database, DBMS[1], Database Management System, Data Warehousing, Multi-level Security, MLS[2], basic robustness, access control, discretionary access control, DAC[3]

1.3 TOE Overview

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

The TOE is a data warehousing product that provides support for Business Intelligence (BI) applications. End users of this product include Chief Information Officers, line-of-business managers, and Chief Executive Officers. The TOE allows these types of users to analyze data trends by processing massive amounts of data at a very high speed. Analysis operations that may take days with other products can take seconds with the TOE architecture.

The TOE is designed for databases ranging from approximately two terabytes to hundreds of terabytes, depending on the model chosen. The TOE uses a proprietary architecture to achieve short query times when compared to traditional distributed data warehousing systems. By combining database, server, and storage components in one design, the product is able to process large amounts of data faster than a traditional data warehousing system. This speed allows the product to perform efficient analytical searches.

1.3.1 Brief Description of the Physical Components of the TOE

The TOE is a database appliance that integrates a database, server, and storage into a single system architecture. The architecture of the TOE database appliance is designed for query speed. Specifically, the TOE architecture is designed to allow efficient, ad-hoc querying of large amounts of data. The TOE employs a technology called Asymmetric Massively Parallel Processing (AMPP) which combines both Symmetric Multiprocessing (SMP) and Massive Parallel Processing (MPP) architectures. Data needing high-level computing at slower speeds can make

[1] DBMS - Database Management System
[2] MLS - Multi-level Security
[3] DAC - Discretionary Access Control

use of SMP, and data needing more speed can make use of MPP. This design enables efficient, high-speed loading and ad-hoc querying of large amounts of data.

In a typical deployment of the TOE, data would be placed into the TOE from a corporate data source (e.g. an e-commerce transactional database, a corporate customer information database, or a corporate wide data collection system). Typically, end users of this product would then access this data through a custom Business Intelligence (BI) application. This BI application would provide the user with mechanisms to perform queries and analyses on sets of data. The BI application accesses the TOE appliance on behalf of the user through standard ODBC[4], JDBC[5], or OLE-DB[6] interfaces to submit SQL[7] queries to the TOE.

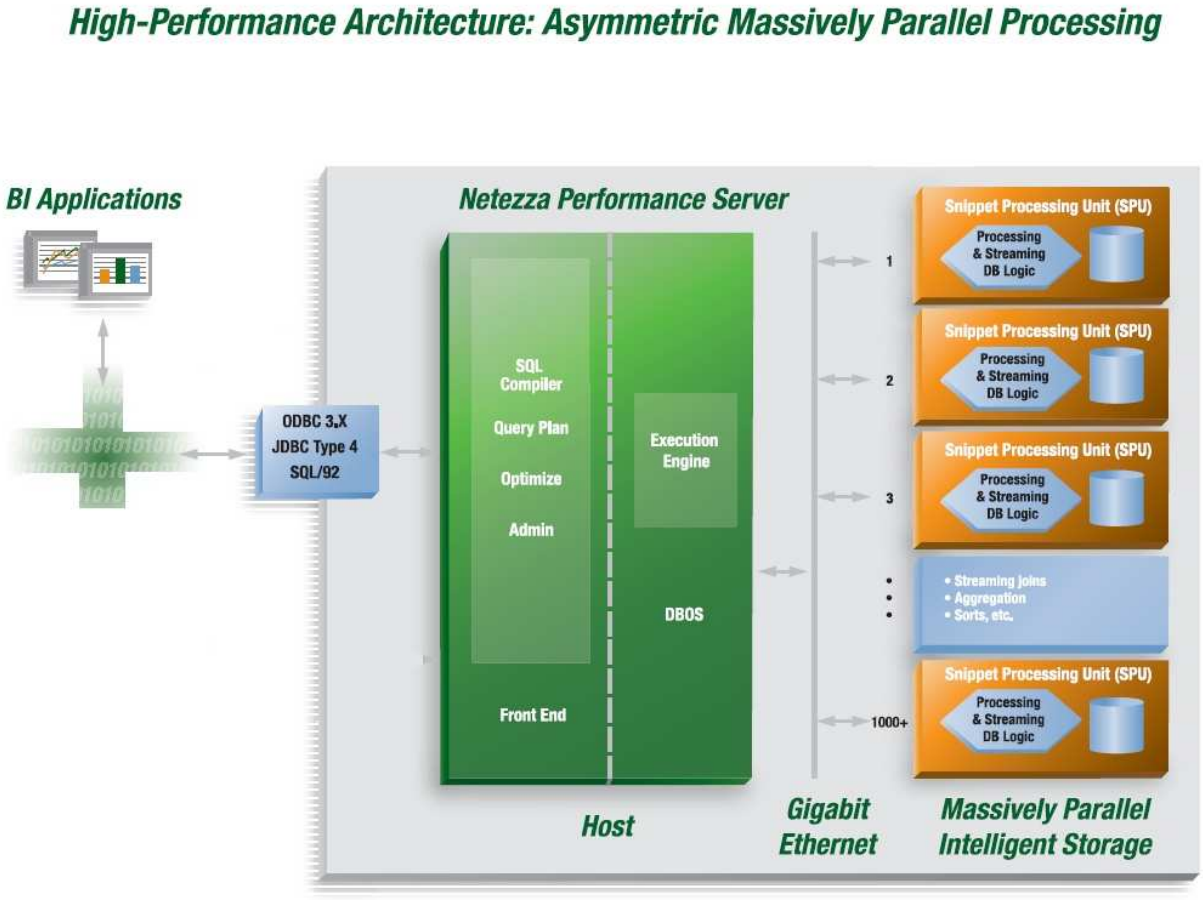
The Netezza Performance Server contains three primary components:

- **Host**
- **Snippet Processing Units (SPUs)**
- **Gigabit Ethernet Switch**

These product components are deployed as shown in Figure 1.

[4] ODBC - Open Database Connectivity
[5] JDBC - Java Database Connectivity
[6] OLE-DB – Object Linking and Embedding Database
[7] SQL - Structured Query Language

Figure 1 - NPS System



1.3.1.1 Host

The Host provides ODBC, JDBC, and OLE-DB connectivity to BI applications and client machines, provides administrative and monitoring functionality, and communicates with individual SPUs for processing queries and storing user data. Before the system offers any services to end users, those users are authenticated by the Host. After successful authentication, a connection is established via either ODBC, JDBC, or OLE-DB. After queries are received by the TOE, they are transformed from standard SQL to a query plan[8]. Next, the query plan is transformed into an optimization plan to achieve the quickest possible results. After the plan has been created it is passed on to an execution engine to manage processing of the query and any transactions that occur on the database. Actual execution of a query is handled by one or more SPUs, with some intermediate and final processing on the host.

All administrative functions of the system are handled by the Host. Input may come from one of three different administrative interfaces. These three interfaces are the NPS Web Admin (a web based administration interface), the nzAdmin, a Windows based Graphical User Interface (GUI), and a Command Line Interface (CLI). Additionally, all audit functions and audit records are managed and stored on the Host. Audits are created for a variety of functions ranging from user access to the start up and shut down of the TOE. As auditable events occur they are written to hard drives on the Host.

[8] Query plan – a set of steps the TOE uses to modify or access information in the database.

1.3.1.2 Snippet Processing Units

Snippet Processing Units (SPUs) are the basic unit of storage and provide query processing, data storage, and data mirroring functionality. SPUs are hardware modules that perform the primitive functions of a query and control all aspects of reading from and writing to a hard drive. Each SPU contains a single hard drive, a dedicated processor, and firmware necessary to process each set of data.

When data is stored by the TOE, that data is distributed among all SPUs. Additionally, each SPU contains a dedicated processor. This allows query operations to occur architecturally close to the storage device. By distributing data across all SPUs and providing separate processors for each storage device, the TOE architecture allows fast, efficient querying of user data. Each SPU also supports the NPS data mirroring scheme. A portion of each disk acts as a primary disk, and a portion acts as a mirror for primary data on another disk. The TOE automatically copies data from a primary to mirror portions of each disk. Mirrors provide fault-tolerance because they provide a redundant and consistent copy of all data stored on each SPU.

1.3.1.3 Gigabit Ethernet Switch

Communication between the Host and the SPUs is accomplished by the third component, a gigabit Ethernet switch. Each TOE rack includes at least one physical switch. The switches connect the SPUs to the Network Interface Cards (NICs) installed on the Host.

1.3.2 Brief Description of the Functionality of the TOE

Database queries generated by BI applications are passed to the TOE where they are processed, the information retrieved, and the results returned to the application. The TOE implements Multi-level Security (MLS) functionality to provide row-level security for sensitive data. MLS allows the TOE to process and store information with different sensitivities, or security levels (such as Top Secret, Secret, Classified, etc.). An MLS system permits simultaneous access by users with different access authorities, and prevents users from obtaining access to information for which they do not have the proper authorization. Individuals with a higher clearance are permitted to access less-sensitive information, and share documents with less-cleared individuals after they have been edited to remove information that the less-cleared individuals are not allowed to see.

1.3.2.1 Multi-level Security

Multi-level Security is a form of access control at the database row level, and is sometimes referred to as Row-level Security or Label Security. Multi-level Security on the TOE requires a multi-dimensional security model, including a security descriptor that defines a level, a set of cohorts, and a set of categories for each data object, or table row. A session security profile for each user or application (also called a “principal”) is also defined by the TOE to have a level, a set of cohorts, and a set of categories.

A level is an indicator of the level of access a given user is granted (e.g., secret, top secret, etc.). Higher levels are more secure; lower levels are less secure. A data object is assigned one level. An individual attempting to access an object must have a level greater than or equal to the level of the object being accessed. The TOE provides two built-in levels as defaults for when the customer does not define any levels for a data object or user: PUBLIC and OMNI. PUBLIC level is the lowest possible level that can be assigned. OMNI level is the highest possible level. The TOE supports approximately 32,000 levels, which are represented by a number in the range 1 to 32766.

A cohort is a hierarchical group assigned by the TOE administrator and is typically used to collect a set of users together. For example, a person’s job title might determine the group to which he is assigned, such as “Executive Team”, or “Engineering”. A cohort is an “any-of” label. A data object may have any number of “any-of” labels. This means that the session security profile of the user attempting to access the data object must match at least one “any-of” label for access to be granted. If the customer has not assigned a cohort to the security descriptor of a data object, it will default to no cohort. If no cohort is assigned then any principal’s security profile will match that security descriptor. Similarly, the cohort of a principal’s security profile will default to no cohort. The TOE allows for more than 64,000 cohorts.

A category is a flat space of tags and is typically used to collect a set of data together. For example, a category might be “cryptology files”, “nuclear weapons”, or “reconnaissance”. A category is an “all-of” label. A data object may have any number of “all-of” labels. This means that the session security profile of the principal attempting to access the data object must match the entire set of “all-of” labels for access to be granted. If a customer has not assigned a category to the security descriptor of a data object, it will default to no category. If no category is assigned, then any principal’s security profile will match that security descriptor. Similarly, the category of a principal’s security profile will default to no category. The TOE supplies at least 64,000 categories.

A session security profile is assigned each time a user or a third-party application logs in to the TOE. Whenever a user or application attempts to access a database row, the TOE will compare the session security profile with the security descriptor for that row to determine whether access will be allowed.

Not all data objects will be required to have Multi-level Security. The customer is able to choose which data objects have Multi-level Security applied to them. In addition, no system catalog has Multi-level Security applied to it; only user objects will have Multi-level Security.

As an extension of the MLS model, the TOE extends the permission system to control the manipulation of the Multi-level Security system, as follows:

Security descriptors and security profiles are also called security labels. Row secure tables are each assigned one of the following permissions:

- LABEL_ACCESS
- LABEL_RESTRICT
- LABEL_EXPAND

LABEL_ACCESS allows the principal attempting to access the row to see the security label (security descriptor) of that row. LABEL_RESTRICT allows the principal to write the security label for a row, but only to make it more restrictive. This means the principal can set a higher level, add one or more categories, or remove one or more cohorts. LABEL_EXPAND allows the principal to write the security label for a row, but only to make it less restrictive. This means the principal can set a lower level, remove one or more categories, or add one or more cohorts.

When a row is inserted into a table, a security label must be specified. The principal must have the LABEL_RESTRICT or LABEL_EXPAND permission to specify a security label on an insert. If no security label is specified by the principal, the label defaults to label of the principal.

When a row is updated, it retains its security label by default. If the principal wishes to set the security label on an update, the principal must have the LABEL_RESTRICT or LABEL_EXPAND permission.

1.3.2.2 Secure Auditing

The TOE also implements secure auditing: the TOE collects query history data and information about authentication attempts and administrative operations, and captures them in one of three types of audit files, depending on the type of event:

- Linux OS log files contain events generated by the OS, such as changes to the system time;
- process-specific disk files on the Host, for processes such as the Backup and Restore Manager, contain text log files;
- an audit database stored on the SPUs contains events that reflect usage of the database by users and administrators.

All logging is performed from the Host, and none from the SPUs.

The audit database is populated by an audit capture server process. The audit capture server process receives and buffers in memory audit data from other Host processes, and then periodically flushes the data from memory to disk. The size of the buffer file is configurable by the administrator. If the audit disk files become full, the audit capture

server process will be unable to write more logs to the files, and will return errors to the processes that are generating the log data. All further activity that requires audit logging will fail until disk space for audit logging has been freed. The TOE will protect the data from tampering by sequencing the data as it is generated by the Host processes. The audit capture server process then further sequences the data as it writes it to disk. In addition, the audit trail is protected by the access control rules in place on the TOE, preventing unauthorized modifications to the audit trail.

After the data is written to disk by the audit capture server process, the audit loader process then validates the two levels of sequencing, and posts the information to a TOE log database in the SPUs on the local machine. This provides data integrity from generation to final storage.

The audit database enforces row-level security. The security label for each row is a combination of two labels: the label of the principal performing the audited action, and the audit categories associated with that principal. The use of the principal's label is to allow only users who are authorized to see the original data to also view the associated audit data. The audit categories prevent the principal responsible for the specific audited action from viewing the associated audit data. This also allows the audit data to be categorized for viewing by more than one auditor.

1.3.2.3 Application Authentication Control

Application control of authentication functionality is another feature of the TOE. The purpose of this functionality is to allow an application to act on behalf of its users. Instead of authenticating every user that accesses the TOE through a third-party application, the TOE will authenticate the application session and grant access to the users of that application as appropriate. The application will authenticate the individual users, and the TOE will enforce the permissions for those users.

When a user identifies and authenticates to an application, and the application identifies and authenticates to the TOE, the user can issue a command to change the 'current user' of the application session. The value of 'current user' in that session changes, and the session takes on the security profile of the 'current user'. When the user issues an end user command, the previous value of the 'current user' is restored.

1.3.3 TOE Environment

Access to a Lightweight Directory Access Protocol (LDAP) server[9] to do user authentication is also a feature of the TOE. Many commercial customers require system integration with LDAP. Many government customers do not want Netezza to provide integration with their authentication service. The common objective of these two requirements is for the TOE to enforce user security, but not to authenticate the end-users. To meet this objective, the TOE implements basic authentication of third-party applications, and then trusts that the application has done authentication on the end users. When LDAP is used, the TOE will support SSL on the connection to the LDAP server.

1.4 TOE Description

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

1.4.1 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 and the constituents of the TOE Environment.

[9] The terms "LDAP server" and "directory server" are used interchangeably throughout this document.

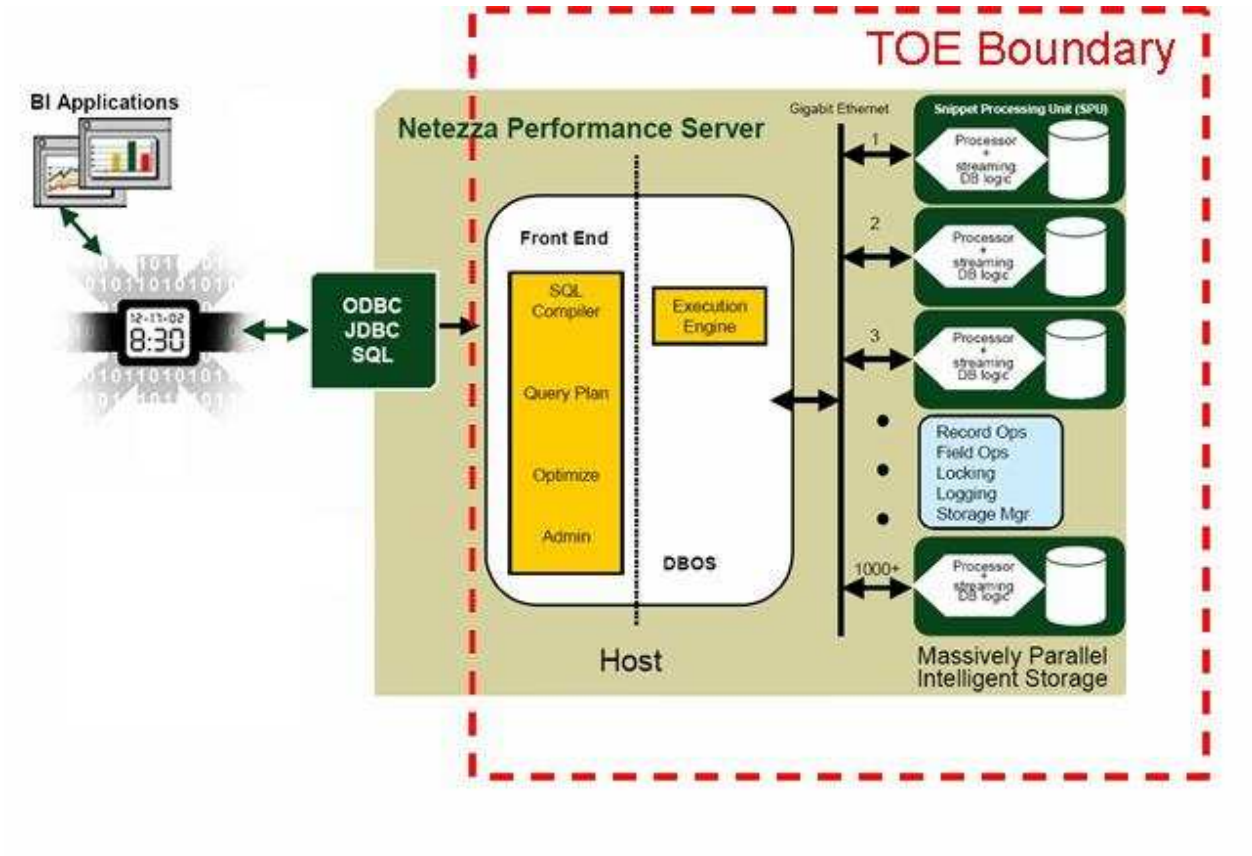


Figure 2 - Physical TOE Boundary

The TOE consists of seven hardware models running the NPS v4.6 software. The seven models are the 5200, 10100, 10200, 10400, 10600, 10800, and 10050.

The three primary physical components that comprise the TOE are:

- **Host:** The Host is the central intelligence component of the TOE architecture. It provides administrative functionality and interfaces with external entities. .
- **SPUs**: Each SPU consists of a hard drive and a processor. This is where low level processing of database queries occurs.
- **Gigabit Ethernet Switch**: The Host and each SPU communicate via an internal network provided by this switch

Other components within the TOE are:

- The CAPP-certified operating system running on the Host: Red Hat Enterprise Linux AS Version 4 Update 4
- Internal audit database
- Power supply
- KVM[10] switch
- Host disk manager
- Host disk(s)

[10] KVM – Keyboard Video Mouse

1.4.1.1 Guidance Documentation

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

- Netezza Advanced Security Administrator's Guide
- Netezza Performance Server Database User's Guide
- Netezza Data Loading Guide
- Netezza Performance Server System Administrator's Guide
- Netezza Performance Server ODBC, JDBC and OLE DB Installation and Configuration Guide
- Netezza Performance Server Release Notes
- Netezza Performance Server Guidance Documentation Supplement

1.4.2 Logical Scope

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
- IT[11] Environment
- Security Management
- Protection of the TSF[12]
- Resource Utilization
- TOE Access

1.4.2.1 Security Audit

One of the primary functions performed by the TOE is the auditing of critical system events. All audit data is stored in one of the various logs residing on the Host or in the audit database residing on the SPUs. Logs are kept which contain the records of regular operations and errors. The system audits numerous functions ranging from software state changes to the start up and shut down of the system. The TOE also records for each event the date and time an event occurred, the type of event, the subject identity or group (if applicable) and the outcome of the event.

Audit logs are available to authorized users to review, and they may perform searches and sorts on the audited data. Administrators may select those events that are to be audited based on the user identity, group identity, event type, object identity, and outcome of the events.

The audit trail is protected from unauthorized deletion and modification. Additionally, an alarm is sent to an administrator if the space available for storing the audit trail exceeds a configured level. No further audits will be recorded if the audit trail becomes full.

1.4.2.2 User Data Protection

User data protection defines how users of the TOE are allowed to perform operations on objects. The TOE is a database and all user data stored by the system is organized within individual database tables. The TOE provides a both a set of discretionary access control rules and a set of label-based access control rules to mediate access to this data. These rights determine the types of operations a user can perform on objects within the database. Additionally, users can be assigned membership to one or more groups. Access rights can then be assigned to

[11] IT – Information Technology
[12] TSF – TOE Security Function

groups, thus providing a richer set of data rights management. In addition, the TOE provides residual information protection upon allocation of resources.

1.4.2.3 Identification and Authentication

All local identification and authentication is managed by the Host component of the TOE. Remote authentication is handled by the Host and an LDAP server for all users except those assigned as 'ADMIN'. 'ADMIN' users must authenticate locally.

All users of the TOE are assigned a username and password. This username and password is then provided during the ODBC, JDBC, or OLE-DB protocol negotiation, or through one of the various management access applications. Users must authenticate themselves before they are granted access to the TOE. There are three possible outcomes for any authentication attempt: the user authentication attempt is correct and the appropriate level of access is granted, the user's attempt is incorrect, but they have not yet submitted enough incorrect attempts to trigger an account lock, or the user has submitted a number of incorrect attempts greater than the number defined by the admin as acceptable, and the account is locked.

Administrators define password reuse, lifetime, and content metrics to ensure passwords are sufficiently strong to prevent unauthorized access to the TOE. Also, the TOE provides only obscured feedback to the user upon entering the password, thereby preventing unauthorized users from obtaining valid account and password information.

Finally, a list of security attributes belonging to individual users is maintained by the TOE, and these are associated with users accessing the TOE. This ensures that all actions taken by a user can be traced to that user.

1.4.2.4 IT Environment

The TOE operating system is compliant with the requirements of the Controlled Access Protection Profile (CAPP), version 1.d, in addition to applicable requirements specified in this Security Target.

1.4.2.5 Security Management

Security Management is provided on the TOE through the NPS Web Admin, the nzAdmin, or the CLI. These applications allow administrators with appropriate privileges to manage the creation and deletion of users and groups. Additionally, this application allows an administrator to assign permissions to users and groups and to revoke permissions from users and groups. This also allows administrators to manage the audit functions of the TOE, thresholds for authentication attempts by users, define limits for resource usage on the TOE by individual users, and specify the number of concurrent sessions each user is permitted to establish at one time.

1.4.2.6 Protection of the TSF

The TOE provides reliable timestamp information for its own use. The time is set through the use of a Network Time Protocol (NTP) client, or manually to the Linux Operating System (OS). From there, other subsystems are able to retrieve the time for inclusion in audit records. Additionally, the TOE ensures that the data remains consistent between parts of the TOE.

1.4.2.7 Resource Utilization

The TOE enforces maximum quotas on the duration of a user's session, the duration of a query, the number of rows that can be returned from a single query, and the percentage of contended system resources that a class of users can use.

1.4.2.8 TOE Access

The TOE allows only a limited number of concurrent TOE sessions for any user, and allows administrators to define when a user may be denied access to the TOE. The TOE will also store and retrieve the date and time of the last successful and unsuccessful attempt to access the TOE by each user.

1.4.3 Physical/Logical Features and Functionality Not Included in the Evaluated Configuration of the TOE

There are no hardware components explicitly excluded from the evaluated configuration. The following features may not be used.

- Password caching is not permitted in the evaluated configuration.
- HP iLO (Hewlett Packard's Integrated Lights-Out) service may not be used.
- User Defined Functions may not be used. These are functions written by end users and installed and executed on the SPUs.

In the evaluated configuration, the following must be implemented:

- Only Authorized Administrators may be given Linux OS accounts.
- The "WITH GRANT OPTION" may only be used when granting privileges to Authorized Administrators. It may not be used when granting privileges to regular users.

2 Conformance Claims

This section provides 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 2 - CC and PP Conformance

Common Criteria (CC) Identification and Conformance	Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 2, September 2007; CC Part 2 extended; CC Part 3 conformant; PP claim; Parts 2 and 3 Interpretations from the Interpreted CEM[13] as of 2008/09/14 were reviewed, and no interpretations apply to the claims made in this ST.
PP Identification	U.S. Government Protection Profile for Database Management Systems in Basic Robustness Environments, Version 1.2
Evaluation Assurance Level	EAL4+ Augmented with Flaw Remediation (ALC_FLR.3)

Note: The U.S. Government Protection Profile for Database Management Systems in Basic Robustness Environments, Version 1.2 contains one IT Environment Security Functional Requirement requiring that the IT Environment be compliant with the Controlled Access Protection Profile or an Operating System Protection Profile at the Basic Level of Robustness or Greater. As v3.1 of the Common Criteria Standard does not permit Environmental SFRs, and the operating system is part of this TOE, this SFR has been changed to a TOE SFR.

Note: The SFR FAU_GEN.1-NIAP-0410 from the PP was changed to the standard FAU_GEN.1, because the wording of the SFR in the PP was the same as the standard CC v3.1 SFR. Therefore, an extended SFR was not necessary to implement the difference in requirements between the extended SFR from the PP and the standard CC version 3.1 SFR.

Note: The SFR FAU_GEN_(EXT).2 from the PP was changed to FAU_GEN.2 and refined in this ST, because an extended SFR was not necessary to implement the difference in requirements between the extended SFR from the PP and the standard CC version 3.1 SFR.

Note: The SFR FAU_SEL.1-NIAP-0407 from the PP was changed to FAU_SEL.1, because an extended SFR was not necessary to implement the difference in requirements between the extended SFR from the PP and the standard CC version 3.1 SFR.

[13] CEM – Common Evaluation Methodology

Note: The SFR FDP_ACF.1-NIAP-0407 from the PP was changed to FDP_ACF.1, because an extended SFR was not necessary to implement the difference in requirements between the extended SFR from the PP and the standard CC version 3.1 SFR.

3 Security Problem Definition

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 assets against which protection is required by the TOE or by the security environment. The threat agents are divided into three 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/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/parameters and physical access to the TOE. (TOE users are, however, assumed not to be willfully hostile to the TOE.)
- TOE developers: 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/parameters and physical access to the TOE. (TOE developers are, however, assumed not to be willfully hostile to the TOE.)

All are assumed to have a low level of motivation. The IT assets requiring protection are the user data saved on or transitioning through the TOE and the hosts on the protected network. Removal, diminution and mitigation of the threats are through the objectives identified in Section 4 - Security Objectives.

The following table of threats is applicable.

Table 3 - Threats

Name	Description
T.ACCIDENTAL_ADMIN_ERROR	An administrator may incorrectly install or configure the TOE resulting in ineffective security mechanisms.
T.MASQUERADE	A user or process may masquerade as another entity in order to gain unauthorized access to data or TOE resources.
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.

Name	Description
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.
T.TSF_COMPROMISE	A user or process may cause, through an unsophisticated attack, TSF data, or executable code 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 administrator to identify and act upon unauthorized actions may occur.
T.RESOURCE	An authenticated database user may consume global database resources, in a way that compromises the ability of other database users to access the DBMS.
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.LBAC	An authorized database user may access labeled information contained within a database without having the authorization to access that information.
T.POOR_DEVELOPMENT_ENVIRONMENT	The TOE's development environment may not protect the TOE and its parts during development and maintenance, may not ensure the TOE meets its SFRs, and may implement ill-defined, inconsistent, or incorrect development tools to develop the TOE.

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. The following OSPs are presumed to be imposed upon the TOE or its operational environment by any organization implementing the TOE in the CC evaluated configuration.

Table 4 - Organizational Security Policies

Name	Description

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.
P.LABEL	Labels can be associated with subjects and with storage objects which are rows within tables: a) A label is composed of a hierarchical level (classification), a set of non-hierarchic categories, and a set of hierarchic groups, as determined by the organization that owns the information stored in the database. b) A storage object label reflects the sensitivity of the information stored in the object. C) A subject label reflects the authorization of the subject to access the organization's labeled information according to defined access rules.
P.INFOFLOW	Information flow from entity A to entity B shall be permitted only if it does not result in a subject being able to observe labeled information that the subject is not authorized to see.
P.OS_PP_VALIDATED	The underlying OS has been validated against an NSA-sponsored OS PP of at least Basic Robustness.

Note regarding P.ACCOUNTABILITY: The TSF will record the process id of the client process and the IP address of the client machine in the audit trail. These identify the application that performs a given action. All processes run as "logical users" in the TOE. Therefore, in this context, a "user" is a user object known to the database system.

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. The following specific conditions are required to ensure the security of the TOE and are assumed to exist in an environment where this TOE is employed.

Table 5 - Assumptions

Name	Description
A.NO_EVIL	Sites using the TOE shall ensure that authorized 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 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.MIDTIER	To ensure accountability in multi-tier environments, any middle-tiers will pass the original client ID through to the TOE.
A.DIR_PROT	The directory server used by the TOE provides protection mechanisms against unauthorized access to TSF data stored in the directory. This includes the assumptions that queries are properly authenticated, that the TSF data stored in the directory is protected by the access control mechanisms of the directory server, that the TSF data in the directory server is properly managed by the administrative personnel, and that the directory server as well as its network connections are physically and logically protected from access and interference by unauthorized persons.
A.DIR_MGMT	The information about users stored in the directory (password verifier, password policy, and privileges) is managed correctly by authorized personnel.
A.COM_PROT	Internal TSF communication as well as communication between the TOE and the directory server are protected from unauthorized access to the transmitted data and ensure that the communication peers are the intended ones.

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, as well as providing a mapping of the objectives to the threats, OSPs, and assumptions included in the security problem definition. This mapping also provides rationale for how the threats, OSPs, and assumptions are effectively and fully addressed by the security objectives.

4.1 Security Objectives for the TOE

The specific security objectives for the TOE are as follows:

Table 6 - 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 isolate administrative actions.
O.AUDIT_GENERATION	The TOE will provide the capability to detect and create records of security relevant events associated with users.
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.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 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

Name	Description
	the TSF satisfies some of its security functional 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.RESIDUAL_INFORMATION	The TOE will ensure that any information contained in a protected resource within its Scope of Control is not released when the resource is allocated.
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.
O.RESOURCE	The TOE must provide the means of controlling the consumption of database resources by authorized users of the TOE.
O.AUDIT_REVIEW	The TOE must provide the means of reviewing the audit log entries allowing users with the required access rights to the audit log to evaluate the audit log entries.
O.AUDIT_PROTECTION	The TOE will provide the capability to protect audit information.
O.ACCESS_LBAC	The TOE must provide the ability for labels to be associated with subjects and database objects in accordance with the P.LABEL security policy. For entities that have been associated with labels, the TOE must use these labels as a basis for implementing an information flow control policy in accordance with the P.INFOFLOW policy.
O.DEVELOPMENT_ENVIRONMENT	The TOE's development environment will protect the TOE and its parts during development and maintenance, ensure that the TOE meets its SFRs, and prevent ill-defined, inconsistent, or incorrect development tools from being used to develop the TOE.
O.OS_PP_VALIDATED	The underlying OS has been validated against an NSA -sponsored OS PP of at least Basic Robustness.

4.2 Security Objectives for the Operational Environment

4.2.1 IT Security Objectives

The following IT security objectives are to be satisfied by the environment:

Table 7 - IT Security Objectives

Name	Description
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.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.
OE.DIR_CONTROL	The directory server must provide access control mechanisms to prohibit unauthorized access to directory entries. The directory server must authenticate users before it allows them to access TSF data stored in the directory.
OE.COM_PROT	The environment must provide protection mechanisms that prohibit unauthorized access to data the TOE transfers over communication links. This applies to data the TOE transmits to another part of itself as well as data exchanged between the TOE and the external directory server. This protection may be provided by physical protection, logical protection, or a combination of both.

4.2.2 Non-IT Security Objectives

The following non-IT environment security objectives 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 8 - Non-IT Security Objectives

Name	Description
OE.NO_EVIL	Sites using the TOE shall ensure that authorized administrators are non-hostile, appropriately trained, and follow all administrator guidance.
OE.USERS	Those responsible for the TOE must ensure that users are assigned label authorizations and policy privileges commensurate with the degree of trust placed in them by the organization that owns, or is responsible for, the information processed by or stored in the TOE.

5 Extended Components Definition

This section defines the extended Security Functional Requirements (SFRs) and extended Security Assurance Requirements (SARs) met by the TOE. These requirements are presented following the conventions identified in Section 6.1.1.

5.1 Extended TOE Security Functional Components

This section specifies the extended SFRs for the TOE. The extended SFRs are organized by class. Table 9 identifies all extended SFRs implemented by the TOE

Table 9 - Extended TOE Security Functional Requirements

Name	Description
FIT_PPC_(EXT).1	IT environment protection profile compliance
FMT_MSA_(EXT).3	Static attribute initialisation
FPT_TRC_(EXT).1	Internal TSF consistency
FTA_TAH_(EXT).1	TOE access history

5.1.1 Class FIT: IT Environment

Note: The name of this extended requirement from the PP (FIT_PPC_(EXT).1) is misleading, as the Operating System is part of the TOE, not the IT Environment.

This class specifies functional requirements for the IT environment. The extended component FIT_PPC_(EXT).1: IT Environment Protection Profile Compliance was modeled after the CC component FDP_ACC.1: Subset access control. This component was originally intended to address conformance to a protection profile by an operating system in the IT environment. However, the TOE boundary for this ST includes the operating system, so this SFR has been changed to a TOE SFR.

5.1.1.1 IT environment protection profile compliance (FIT_PPC_(EXT))

Family Behaviour

This family identifies the protection profile compliance claim for the specified component.

Component Leveling

Figure 3 – IT environment protection profile compliance family decomposition

FIT_PPC_(EXT).1 requires that the stated component comply with the specified protection profile.

Management: FIT_PPC_(EXT).1

There are no management activities foreseen.

Audit: FIT_PPC_(EXT).1

There are no auditable events foreseen.

FIT_PPC_(EXT).1 IT environment protection profile compliance

Hierarchical to: No other components

Dependencies: No dependencies

FIT_PPC_(EXT).1.1

The [assignment: *component*] shall be compliant with the requirements of [assignment: *protection profile*].

5.1.2 Class FMT: Security Management

Security management is intended to specify the management of several aspects of the TSF: security attributes, TSF data, and functions. The different management roles and their interactions, such as separation of capability, can be specified. The extended component FMT_MSA_(EXT).3: Static attribute initialisation was modeled after the CC component FMT_MSA.3: Static attribute initialisation.

5.1.2.1 Management of security attributes (FMT_MSA_(EXT))

Family Behaviour

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

Component Leveling

Figure 4 - Management of security attributes family decomposition

FMT_MSA_(EXT).3 Static attribute initialisation specifies that the TSF shall provide restrictive or permissive default values for security attributes that are used to enforce the named SFP[14].

Management: FMT_MSA_(EXT).3

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

a) Management of the group of database roles that can specify initial values;
b) Management of the permissive or restrictive setting of default values for a given SFP.

Audit: FMT_MSA_(EXT).3

There are no auditable events foreseen.

FMT_MSA_(EXT).3 Static attribute initialisation

Hierarchical to: No other components

Dependencies: FMT_MSA.1 Management of security attributes

FMT_SMR.1 Security roles

FMT_MSA_(EXT).3.1

The TSF shall enforce the [assignment: a*ccess control SFP, information flow control SFP*] to provide [selection, choose one of: *restrictive, permissive, [assignment: other property]*] default values for security attributes that are used to enforce the SFP.

[14] SFP – Security Functional Policy

5.1.3 Class FPT: Protection of the TSF

This class contains families of functional requirements that relate to the integrity and management of the mechanisms that constitute the TSF and to the integrity of TSF data. The extended component FPT_TRC_(EXT).1: Internal TSF consistency was modeled after the CC component FPT_TRC.1: Internal TSF consistency.

5.1.3.1 Internal TSF consistency (FPT_TRC_(EXT))

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.

Component Leveling

Figure 5 - Internal TSF consistency family decomposition

This family consists of only one component, FPT_TRC_(EXT).1 Internal TSF consistency, which requires that the TSF ensure the consistency of TSF data that is replicated in multiple locations.

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:

a) Minimal: restoring consistency.

FPT_TRC_(EXT).1 Internal TSF consistency

Hierarchical to: No other components

Dependencies: No dependencies

FPT_TRC.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.

5.1.4 Class FTA: TOE access

This class specifies functional requirements for controlling the establishment of a user's session. The extended component FTA_TAH_(EXT).1: TOE access history was modeled after the CC component FTA_TAH.1: TOE access history.

5.1.4.1 TOE access history (FTA_TAH_(EXT))

Family Behaviour

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

Component Leveling

Figure 6 - TOE access history family decomposition

FTA_TAH_(EXT).1 TOE access history, provides the requirement for a TOE to display information related to previous attempts to establish a session.

Management: FTA_TAH_(EXT).1

There are no management activities foreseen.

Audit: FTA_TAH_(EXT).1

There are no auditable events foreseen.

FTA_TAH_(EXT).1 TOE access history

Hierarchical to: No other components

Dependencies: No dependencies

FTA_TAH_(EXT).1.1

Upon successful session establishment, the TSF shall store and retrieve the [selection: *date*, *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 [selection: *date*, *time*] of the last unsuccessful attempt to session establishment and the number of unsuccessful attempts since the last successful session establishment.

5.2 Extended TOE Security Assurance Components

There are no extended SARs identified for this ST.

6 Security Requirements

This section defines the Security Functional Requirements (SFRs) and Security Assurance Requirements (SARs) met by the TOE. These requirements are presented following the conventions identified in Section 6.1.1.

6.1.1 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 Parts 2 and 3 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 italicized 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 end of the short name.
- Iterations are identified by appending a letter in parentheses following the component title. For example, FAU_GEN.1(a) Audit Data Generation would be the first iteration and FAU_GEN.1(b) Audit Data Generation would be the second iteration.

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	Audit Data Generation	✓	✓	✓	
FAU_GEN.2	User identity association			✓	
FAU_SAR.1	Audit review		✓		
FAU_SAR.2	Restricted audit review				
FAU_SAR.3	Selectable audit review		✓		
FAU_SEL.1	Selective audit	✓	✓		
FAU_STG.1	Protected audit trail storage	✓			

Name	Description	S	A	R	I
FAU_STG.3	Action in case of possible audit data loss		✓		
FAU_STG.4	Prevention of audit data loss	✓	✓	✓	
FDP_ACC.1	Subset access control		✓		
FDP_ACF.1	Security attribute based access control		✓	✓	
FDP_IFC.1	Subset information flow control		✓		
FDP_IFF.2	Hierarchical security attributes		✓	✓	
FDP_RIP.1	Subset residual information protection	✓	✓		
FIA_AFL.1	Authentication failure handling	✓	✓		
FIA_ATD.1	User attribute definition		✓		
FIA_SOS.1	Verification of secrets		✓	✓	
FIA_UAU.1	Timing of authentication		✓	✓	
FIA_UAU.7	Protected authentication feedback		✓		
FIA_UID.1	Timing of identification		✓		
FIA_USB.1	User-subject binding		✓		
FIT_PPC_(EXT).1	IT Environment		✓		
FMT_MOF.1(a)	Management of security functions behaviour	✓	✓		✓
FMT_MOF.1(b)	Management of security functions behaviour	✓	✓		✓
FMT_MSA.1(a)	Management of security attributes	✓	✓	✓	✓
FMT_MSA.1(b)	Management of security attributes	✓	✓		✓

Name	Description	S	A	R	I
FMT_MSA_(EXT).3	Static attribute initialisation	✓	✓		
FMT_MSA.3	Static attribute initialisation	✓	✓	✓	
FMT_MTD.1(a)	Management of TSF data	✓	✓		✓
FMT_MTD.1(b)	Management of TSF data	✓	✓		✓
FMT_REV.1(a)	Revocation	✓	✓	✓	✓
FMT_REV.1(b)	Revocation	✓	✓	✓	✓
FMT_SMF.1	Specification of management functions		✓		
FMT_SMR.1	Security roles		✓		
FPT_STM.1	Reliable time stamps				
FPT_TRC_(EXT).1	Internal TSF consistency				
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 Audit data generation

Hierarchical to: No other components.

FAU_GEN.1.1

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

- Start-up and shutdown of the audit functions;
- All auditable events, for the [*minimum*] level of audit **listed in Table 11 below**; and
- [*Start-up and shutdown of the DBMS;*
- *Use of special permissions (e.g., those often used by authorized administrators to circumvent access control policies); and*
- *Events at a minimal level of audit introduced by the inclusion of additional SFRs determined by the ST author; events commensurate with a minimal level of audit introduced by the inclusion of extended requirements determined by the ST author*].

Note: The audit functions run as part of the DBMS functionality. For a normal shutdown, the audited state change will reflect the shutdown of the audit functions and the database.

Note: There is no distinction between "special permissions" and "normal permissions" used by administrators in the TOE. All administrator actions are captured in the audit log.

Table 11 - Auditable Events

Component	Auditable Event	Additional Audit Record Contents
FAU_GEN.1	None	None
FAU_GEN.2	None	None
FAU_SAR.1	Reading of information from the DATABASE audit records	None
FAU_SAR.2	Unsuccessful attempts to read information from the audit records	None
FAU_SAR.3	None	None
FAU_SEL.1	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

Component	Auditable Event	Additional Audit Record Contents
FAU_STG.1	None	None
FAU_STG.3	Actions taken due to exceeding of a threshold	None
FAU_STG.4	Actions taken due to audit storage failure	None
FDP_ACC.1	None	None
FDP_ACF.1	Successful requests to perform an operation on an object covered by the SFP	The identity of the subject performing the operation
FDP_IFC.1	None	None
FDP_IFF.2	All decisions on requests for information flow	None
FDP_RIP.1	None	None
FIA_AFL.1	The reaching of the threshold for the unsuccessful authentication attempts and the actions (e.g., disabling of a terminal) taken and the subsequent, if appropriate, restoration to the normal state (e.g., re-enabling of a terminal)	None
FIA_ATD.1	None	None
FIA_SOS.1	Successful and unsuccessful attempts to change a user's password	None
FIA_UAU.1	All use of the DATABASE user authentication mechanism, including success or failure of the authentication attempt	None
FIA_UAU.7	None	None
FIA_UID.1	All use of the DATABASE user identification mechanism, including the DATABASE user identity provided	None
FIA_USB.1	Success and failure of binding of DATABASE user security attributes to a DATABASE subject (e.g., success and failure to create a DATABASE subject)	None
FMT_MOF.1(a)	None	None

Component	Auditable Event	Additional Audit Record Contents
FMT_MOF.1(b)	All modifications in the behaviour of the functions in the TSF	None
FMT_MSA.1(a)	All modifications of the values of security attributes	None
FMT_MSA.1(b)	All modifications to security labels	None
FMT_MSA_(EXT).3	None	None
FMT_MSA.3	Modifications of the default setting of permissive or restrictive DATABASE OBJECT LABEL rules All modifications of the initial value of security attributes	None
FMT_MTD.1(a)	All modifications to the values of TSF data	The new value of the TSF data
FMT_MTD.1(b)	All modifications to the values of TSF data	None
FMT_REV.1(a)	Unsuccessful revocation of security attributes	Identity of individual attempting to revoke security attributes
FMT_REV.1(b)	Unsuccessful revocation of security attributes	Identity of individual attempting to revoke security attributes
FMT_SMF.1	Use of the 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_STM.1	Changes to the time[15]	None
FPT_TRC_(EXT).1	Restoring consistency	None
FRU_RSA.1	All attempted uses of the DATABASE resource allocation functions for resources that are under control of the TSF	None
FTA_MCS.1	Rejection of a new session based on the limitation of multiple concurrent sessions	None

[15] Changes to the time are captured in the system log, not in the audit database.

Component	Auditable Event	Additional Audit Record Contents
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
FIT_PPC_(EXT).1	As defined by CAPP or other operating system protection profile at the basic level of robustness or greater.	As defined by CAPP or other operating system protection profile at the basic level of robustness or greater.

FAU_GEN.1.2

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

- Date and time of the event, type of event, subject identity (if applicable), and the outcome (success or failure) of the event; and
- For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [*information specified in column 3 of Table 11 above*].

Dependencies: FPT_STM.1 Reliable time stamps

FAU_GEN.2 User identity association

Hierarchical to: No other components.

FAU_GEN.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

FAU_SAR.1 Audit review

Hierarchical to: No other components.

FAU_SAR.1.1

The TSF shall provide [*users with read access to the audit records*] with the capability to read [*all database audit information to which they have access*] 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 [*searches and sorts*] of audit data based on [*the values of audit data fields*].

Dependencies: FAU_SAR.1 Audit review

FAU_SEL.1 Selective audit

Hierarchical to: No other components.

FAU_SEL.1.1

The TSF shall ~~be able~~ **allow only the administrator** to select the set of audited events from the set of auditable 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;*

g) *no additional criteria;*

].

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

Note: The term "object" in this case refers to the entire database.

Note: Every auditable event in the TOE produces two audit records: a prologue and an epilogue. The prologue audit contains the data for an event that has not yet executed, such as a query, and, as such does not contain an outcome. The epilogue contains the data for the event after it has executed, and includes the outcome of the event. The prologue and epilogue records are linked, and can be viewed as a set. Administrators can choose to audit only successful or failed events, but will be able to prevent only the (successful or failed) epilogue audit from occurring. This will result in the recording of single prologue events (without an associated epilogue).

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

FAU_STG.3 Action in case of possible audit data loss

Hierarchical to: No other components.

FAU_STG.3.1

The TSF shall take [*generate an alarm to the authorized administrator*] if the audit trail exceeds [*the administrator-configurable limit on amount of space used to buffer the audit trail and the limit on amount of space available for storing audit data*].

Dependencies: FAU_STG.1 Protected audit trail storage

FAU_STG.4 Prevention of audit data loss

Hierarchical to: FAU_STG.3 Action in case of possible audit data loss

FAU_STG.4.1

The TSF shall [*prevent audited events, except those taken by the authorised user with special rights*] and [*actions defined by FAU_STG.3*] if the audit trail is full.

Dependencies: FAU_STG.1 Protected audit trail storage

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 [*Discretionary Access Control SFP*] on [*all subjects, all DMBS-controlled objects, and all operations among them*].

Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACF.1 Security attribute based access control

Hierarchical to: No other components.

FDP_ACF.1.1

The TSF shall enforce the [*Discretionary Access Control SFP*] to objects based on the following:

[

a) *the authorized user identity and/or group membership associated with a subject;*

b) *access operations implemented for DBMS-controlled objects;*

c) *object identity;*

d) subject privileges; and

e) object access privileges

].

FDP_ACF.1.2

The TSF shall enforce the following rules to determine if an operation among controlled subjects and **DBMS-**controlled objects is allowed:

[

The Discretionary Access Control policy 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, deny access;

b) If the requested mode of access is permitted to that authorized user, permit access;

c) If the requested mode of access is denied to every group of which the authorized user is a member, deny access;

d) If the requested mode of access is permitted to any group of which the authorized user is a member, grant access;

e) Else, deny access

].

FDP_ACF.1.3

The TSF shall explicitly authorise access of subjects to **DBMS-controlled** objects based on the following additional rules: [

If the database subject has a database administrative privilege[16] *to override the database object access controls for the requested access to the database object, then the requested access is allowed;*

].

FDP_ACF.1.4

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

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

Note: The Label-based Access Control SFP is also to be applied to database subjects, objects, and operations as specified in SFR FDP_IFC.1.1 and SFRs FDP_IFF.2.1 to FDP_IFF.2.6. The Label-based Access Control SFP applies controls that are additional to the Discretionary Access Control SFP.

FDP_IFC.1 Subset information flow control

Hierarchical to: No other components.

FDP_IFC.1.1

The TSF shall enforce the [*Label-based Access Control SFP*] on

[

a) Database subjects;

b) Labeled database objects;

c) All permitted operations on labeled objects by a database subject covered by the SFP

].

Dependencies: FDP_IFF.1 Simple security attributes

[16] Database administrative privilege refers to the privileges granted to the owner of a given database. This is one type of "administrator privilege" within the NPS system. Possible administrator privileges include: Backup, Create Database, Create Group, Create Table, Create User, Create View, Reclaim, Restore, and Manage System, among others. For further explanation of these privileges, please see the *Netezza Performance Server System Administrator's Guide, Document Number: 20282-11 Rev. 1, Software Release: 4.6.x, Revised: February 3, 2009*, Chapter 8.

FDP_IFF.2 Hierarchical security attributes

Hierarchical to: FDP_IFF.1 Simple security attributes

FDP_IFF.2.1

The TSF shall enforce the [*Label-based Access Control SFP*] based on the following types of subject and information security attributes:

[

a) Database subject security labels; and

b) Security labels of the database object containing the information

].

Note: Security labels shall include a hierarchic classification level and a (possibly empty) set of non-hierarchic categories and a (possibly empty) set of hierarchic groups (cohorts). An object is to have exactly one label.

FDP_IFF.2.2

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

[

a) A database subject may observe the contents of a database object only if, for every NPS policy that applies to the object:

The session label of the database subject dominates[17] *the label of the database object;*

And

b) A database subject may modify a database object only if, for every NPS policy that applies to the object:

The subject has a session label that dominates the row label, and the subject has UPDATE permissions on the table,

And

(

the subject is not changing the security label of the object, or changing it to an equal value, and is altering other columns;

Or

the subject has LABEL_RESTRICT permission, and the subject is changing the security label to a label that is greater than the existing label;

[17] This means that the principal has a level of greater or equal to the object (row) level, and the principal has the same or a superset of the categories of the object, and the principal has at least one cohort in common with the object.

Or

the subject has LABEL_EXPAND permission, and the subject is changing the security label to a label that is less than the existing label;

Or

the subject has LABEL_RESTRICT permission, and the subject has LABEL_EXPAND permission, and the subject is changing the security label to a label that is mixed: parts greater, parts less

).

]

Note: NPS policies assigned to objects shall specify which controls are to be applied when a subject attempts to access an object.

FDP_IFF.2.3

The TSF shall enforce the

[

capability to execute a stored procedure, function, or package at the executing user's current session label and with the set of label-based access control privileges given to the stored procedure, function, or package.

].

FDP_IFF.2.4

The TSF shall explicitly authorise an information flow based on the following rules:

[*none*].

FDP_IFF.2.5

The TSF shall explicitly deny an information flow based on the following rules:

[

None

].

FDP_IFF.2.6

The TSF shall enforce the following relationships for any two valid information flow control security attributes **(security labels)**:

- There exists an ordering function that, given two valid **security labels**, determines if the **security labels** are equal, if one **security label** is greater than the other, or if the **security labels** are incomparable; and

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

- There exists a "greatest lower bound" in the set of **security labels**, such that, given any two valid **security labels**, there is a valid **security labels** that is not greater than the two valid **security labels**.

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

Dependencies: FDP_IFC.1 Subset information flow control
FMT_MSA.3 Static attribute initialisation

FDP_RIP.1 Subset residual information protection

Hierarchical to: No other components.

FDP_RIP.1.1

The TSF shall ensure that any previous information content of a resource is made unavailable upon the [*allocation of the resource to*] the following objects: [*schema[18] objects (including non-schema[19] objects that are stored in the sys schema[20])*].

Dependencies: No dependencies

[18] Schema objects are objects stored in databases. In the TOE, these include tables, views, sequences, user-defined functions, user-defined aggregates, row secure tables, etc.
[19] Non-schema objects are global metadata objects, such as users, groups, categories, cohorts, and labels.
[20] Sys schema refers to the SYSTEM database where the metadata describing the schema objects and non-schema objects is stored.

6.2.3 Class FIA: Identification and Authentication

FIA_AFL.1 Authentication failure handling

Hierarchical to: No other components.

FIA_AFL.1.1

The TSF shall detect when [*an administrator configurable positive integer within [1 to infinite]*] unsuccessful authentication attempts occur related to [*the unsuccessful authentication attempts since the last successful authentication to the Netezza Performance Servers*].

FIA_AFL.1.2

When the defined number of unsuccessful authentication attempts has been [*met*], the TSF shall [*lock the user account until it is re-enabled by the administrator*].

Dependencies: FIA_UAU.1 Timing of 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:

[

a) Database user identifier and group memberships[21]*; and*

b) **Permissions granted to the user as defined by** *security-relevant database roles; and*

c) Object or system privilege; and

d) For each NPS Policy for which the user has authorization:

a. *a level indicator;*

b. *a (possibly empty) set of authorised categories;*

c. *a (possibly empty) set of authorised groups;*

d. *an initial session label;*

e. *a (possibly empty) set of label-based access control policy privileges.*

].

Dependencies: No dependencies

[21] All users are always members of the PUBLIC group. Administrators can add users to other groups and assign users to a resource group. All users have a user id.

Note: For this TOE, "group memberships" are logically equivalent to "roles". Groups provide the assignment of privileges to individual users.

FIA_SOS.1 Verification of secrets

Hierarchical to: No other components.

FIA_SOS.1.1

The TSF shall provide a mechanism to verify that secrets **(passwords for local users)** meet

[

Reuse, lifetime, and content metrics as defined by an authorized administrative user

].

Dependencies: No dependencies

FIA_UAU.1 Timing of authentication

Hierarchical to: No other components.

FIA_UAU.1.1

The TSF shall allow [*user identification and password entry*] on behalf of the user to be performed before the user is authenticated.

FIA_UAU.1.2

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

Dependencies: FIA_UID.1 Timing of identification

Note: Identification and Authentication claims listed in this ST apply to both administrative users and database users. This functionality is provided by the database system, not the operating system. The operating system also performs Identification and Authentication for some administrative tasks. SFRs covering this functionality can be found in the Controlled Access Protection Profile (CAPP).

FIA_UAU.7 Protected authentication feedback

Hierarchical to: No other components.

FIA_UAU.7.1

The TSF shall provide only [*obscured feedback*] to the user while the authentication is in progress.

Dependencies: FIA_UAU.1 Timing of authentication

FIA_UID.1 Timing of identification

Hierarchical to: No other components.

FIA_UID.1.1

The TSF shall allow [*user id entry and password entry*] on behalf of the user to be performed before the user is identified.

FIA_UID.1.2

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

Dependencies: No dependencies

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:
[

a) User identifier, object or system privileges, and groups;

b) Level indicator, set of authorized categories. set of authorized groups, initial session label, and policy privileges

].

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:

[

a) Once a user has been successfully identified and authenticated at the start of a session with the TSF, the user's identifier is accessible throughout that session;

b) An object or system privilege is effective at the start of a user session if it was previously granted to the user (and not subsequently revoked) directly, via the public user group, or granted to a user group in which the user is a member;

c) An NPS policy privilege will be effective for the policy in an active user session immediately upon a policy change;

d) At the start of a user session, the session label and default row label for each applicable NPS policy are set to the user's initial session label and initial default row label attributes.

].

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:

[

a) If an object of system privilege applying to a user is granted or revoked while the user has a current session with the TSF, this change applies to the set of locally managed privileges effective the next time the user logs on. This rule applies to privileges granted to the user directly or via the public user group;

b) If a user executes a view or a program unit owned by another user that was created with "definer's rights", the privileges of the owning user are effective during the execution of the view or program unit;

c) A local user can change the password associated with that user if the new password complies with the configurable controls included in the password management information that applies to the user;

d) An NPS Policy privilege changed during a session only becomes effective at the start of the next user session;

e) During the execution of a stored procedure, function, or package, the security label of the stored procedure, function, or package is effective. If the stored procedure, function, or package does not have a defined security label, then the user's session label is effective;

f) During the execution of a trigger, the session label and the policy privileges of the user that invoked the trigger are effective

].

Dependencies: FIA_ATD.1 User Attribute Definition

6.2.4 Class FIT: IT Environment

FIT_PPC_(EXT).1 IT Environment Protection Profile Compliance

Hierarchical to: No other components.

FIT_PPC_(EXT).1.1

The [*TOE Operating System*] shall be compliant with the requirements of [*the Controlled Access Protection Profile or an Operating System Protection Profile at the Basic Level of Robustness or greater*].

Dependencies: No dependencies

Note: This requirement can be met by providing evidence (e.g., certificate) that the underlying operating system is compliant with the Controlled Access Protection Profile or with a protection profile at the Basic Level of Robustness or greater.

Note: The name of this extended requirement from the PP is misleading, as the Operating System is part of the TOE, not the IT Environment.

6.2.5 Class FMT: Security Management

FMT_MOF.1(a) Management of security functions behaviour

Hierarchical to: No other components.

FMT_MOF.1.1(a)

The TSF shall restrict the ability to [*disable, 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_MOF.1(b) Management of security functions behaviour

Hierarchical to: No other components.

FMT_MOF.1.1(b)

The TSF shall restrict the ability to [*modify the behaviour of*] the functions [*Label-based Access Control SFP functions*] to [*authorized administrative users*].

Dependencies: FMT_SMF.1 Specification of management functions
FMT_SMR.1 Security roles

Note: The term "modify the behaviour of" in this SFR refers to the ability to define and change the rules and management activities for the Label-based Access Control SFP .

FMT_MSA.1(a) Management of security attributes

Hierarchical to: No other components.

FMT_MSA.1.1(a)

The TSF shall enforce the [*Discretionary Access Control SFP*] to restrict the ability to [*manage*] **all** the security attributes [*stored and managed by the TSF*] to [*authorized administrators*].

Dependencies: [FDP_ACC.1 Subset access control or
FDP_IFC.1 Subset information flow control]
FMT_SMF.1 Specification of management functions
FMT_SMR.1 Security roles

Note: The term "manage" in this SFR refers to the ability to configure the values of the security attributes defined by FIA_ATD.1.

FMT_MSA.1(b) Management of security attributes

Hierarchical to: No other components.

FMT_MSA.1.1(b)

The TSF shall enforce the [*Label-based Access Control SFP*] to restrict the ability to [*modify*] the security attributes [*labels and privileges*] to [*authorized users*].

Dependencies: [FDP_ACC.1 Subset access control or
FDP_IFC.1 Subset information flow control]
FMT_SMF.1 Specification of management functions
FMT_SMR.1 Security roles

Note: The term "modify" in this SFR refers to the ability to define and change the information control security attributes used in controlling access to database objects.

FMT_MSA_(EXT).3 Static attribute initialisation

Hierarchical to: No other components.

FMT_MSA_EXP.3.1

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

Dependencies: FMT_MSA.1 Management of security attributes
FMT_SMR.1 Security roles

Note: This requirement applies to new container objects at the top-level (e.g., tables). When lower-level objects are created (e.g., rows, cells), these may inherit the permissions of the top-level objects by default. In other words, the permissions of the 'child' objects can take the permissions of the 'parent' objects by default.

Note: The security attributes referred to by the SFR are object permissions. The default value of these attributes must satisfy specific criteria, and not be blank. If the object permissions default to some value, the value must be restrictive, and cannot be overridden.

FMT_MSA.3 Static attribute initialisation

Hierarchical to: No other components.

FMT_MSA.3.1

The TSF shall enforce the [*Label-based Access Control SFP*] to provide [*no*] default values for **database object** security attributes that are used to enforce the SFP.

FMT_MSA.3.2

The TSF shall allow the [*no database users*] to specify alternative initial values to override the default values **for Label-based Access Control security attributes** when ~~a~~**n database** object ~~or information~~ is created.

Dependencies: **FMT_MSA.1 Management of security attributes**
FMT_SMR.1 Security roles

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

FMT_MTD.1(a) Management of TSF data

Hierarchical to: No other components.

FMT_MTD.1.1(a)

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_MTD.1(b) Management of TSF data

Hierarchical to: No other components.

FMT_MTD.1.1(b)

The TSF shall restrict the ability to [*perform operations on*] the [*TSF data as listed in Table 12 below*] to [*authorized administrators*].

Table 12 - Management Events

Component	Operation	TSF Data
FAU_GEN.1	None	None
FAU_GEN.2	None	None
FAU_SAR.1	Maintenance (deletion, modification, addition)	The group of database users with read access rights to the database audit records
FAU_SAR.2	None	None
FAU_SAR.3	None	None
FAU_SEL.1	Maintenance of the right to view/modify	The database audit events
FAU_STG.1	None	None

Component	Operation	TSF Data
FAU_STG.3	Maintenance	a) The threshold b) actions to be taken in case of imminent audit storage failure
FAU_STG.4	Maintenance (deletion, modification, addition)	Actions to be taken in case of audit storage failure
FDP_ACC.1	None	None
FDP_ACF.1	Managing	The attributes used to make explicit access- or denial-based decisions
FDP_IFC.1	None	None
FDP_IFF.2	Manage	The attributes used to make explicit access- or denial-based decisions
FDP_RIP.1	None	None
FIA_AFL.1	Management	a) The threshold for unsuccessful database authentication attempts b) Actions to be taken in the event of a database authentication failure
FIA_ATD.1	None	None
FIA_SOS.1	Management	The metric used to verify the database secrets
FIA_UAU.1	Manage	a) The database authentication data
FIA_UAU.7	None	None
FIA_UID.1	Management	The user identities
FIA_USB.1	None	None
FIT_PPC_(EXT).1	None	None
FMT_MOF.1(a)	Manage	Auditable events
FMT_MOF.1(b)	Manage	The group of roles that can interact with the Label-based Access Control functions

Component	Operation	TSF Data
FMT_MSA.1(a)	Manage	The group of database roles that can interact with the database security attributes
FMT_MSA.1(b)	Manage	The group of database roles that can interact with the database object labels
FMT_MSA_(EXT).3	Manage	a) The group of database roles that can specify initial values b) The permissive or restrictive setting of default values for the Discretionary Access Control SFP
FMT_MSA.3	None	None
FMT_MTD.1(a)	Include or exclude	Auditable events
FMT_MTD.1(b)	Manage	The group of database roles that can interact with the TSF data
FMT_REV.1(a)	None	None
FMT_REV.1(b)	None	None
FMT_SMF.1	None	None
FMT_SMR.1	Manage	The group of database users that are part of a database role
FPT_STM.1	Manage	The time
FPT_TRC_(EXT).1	None	None
FRU_RSA.1	Specify	Maximum limits for a resource for database groups or individual database users or database subjects by a database administrator
FTA_MCS.1	Manage	The maximum allowed number of concurrent database user database sessions by a database administrator
FTA_TAH_(EXT).1	None	None
FTA_TSE.1	None	None

Dependencies: FMT_ SMF.1 Specification of management functions
FMT_SMR.1 Security roles

FMT_REV.1(a) Revocation

Hierarchical to: No other components.

FMT_REV.1.1(a)

The TSF shall restrict the ability to revoke [*security attributes*] associated with the [*database users*] under the control of the TSF to [*the authorised administrator*].

FMT_REV.1.2(a)

The TSF shall enforce the **following** rules:

[

Revocation of database administrative privileges shall take effect prior to when the database user begins the next database session

].

Dependencies: FMT_SMR.1 Security roles

FMT_REV.1(b) Revocation

Hierarchical to: No other components.

FMT_REV.1.1(b)

The TSF shall restrict the ability to revoke [*security attributes*] associated with the [*objects*] under the control of the TSF to [*the authorized administrator and database users as allowed by the Discretionary Access Control SFP*].

FMT_REV.1.2(b)

The TSF shall enforce the **following** rules:

[

a) Revocation of database object access privileges shall take effect prior to all subsequent attempts to establish access to the database object;

b) Revocation of database administrative privileges shall take effect prior to when the database user begins the next database session

].

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 functions behavior (as outlined in FMT_MOF.1(a) and FMT_MOF.1(b)), Management of security attributes (as outlined in FMT_MSA.1(a), FMT_MSA.1(b), FMT_MSA_(EXT).3, FMT_MSA.3, FMT_REV.1(a), FMT_REV.1(b), and FMT_SMR.1), and Management of TSF data (as outlined in FMT_MTD.1(a), FMT_MTD.1(b))*].

Dependencies: No Dependencies

FMT_SMR.1 Security roles

Hierarchical to: No other components.

FMT_SMR.1.1

The TSF shall maintain the roles:

[

a) authorized administrator (including 'ADMIN' role),

b) database user

].

FMT_SMR.1.2

The TSF shall be able to associate users with roles.

Dependencies: FIA_UID.1 Timing of identification

6.2.6 Class FPT: Protection of the TSF

FPT_STM.1 Reliable time stamps

Hierarchical to: No other components.

FPT_STM.1.1

The TSF shall be able to provide reliable time stamps.

Dependencies: No dependencies

FPT_TRC_(EXT).1 Internal TSF consistency

Hierarchical to: No other components.

FPT_TRC.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

6.2.7 Class FRU: Resource Utilization

FRU_RSA.1 Maximum quotas

Hierarchical to: No other components.

FRU_RSA.1.1

The TSF shall enforce maximum quotas of the following resources:

[

a) Duration of a session;

b) Duration of a query;

c) Number of rows that can be returned from a single query; and

d) Percentage of contended system resources[22] that a class of users[23] can use

]

that [*an individual user*] can use [*over a specified period of time*].

Dependencies: No dependencies

[22] Contended system resources are resources that users share, such as CPU time, disk space, memory, or network bandwidth.

[23] A class of users is a user group that is designated as the resource group for a set of users. Resource groups are used to control the percentage of resources the group can use, and are part of the Workload Management function of the TOE. Users are assigned to resource groups by the system administrator, the owner of the group, or another appropriately-privileged user. For more information about resource groups, please refer to *Netezza Performance Server System Administrator's Guide, Document Number: 20282-11 Rev. 1, Software Release: 4.6.x, Revised: February 3, 2009*, Chapters 1, 5, 10, and 11.

6.2.8 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.

FTA_MCS.1.2

The TSF shall enforce, by default, a limit of [*an administrator-configurable number of*] 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, 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, 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

The TSF shall be able to deny session establishment based on [*attributes that can be set explicitly by authorized administrators, 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 EAL4 augmented with ALC_FLR.3. Table 13 - Assurance Requirements summarizes the requirements.

Table 13 - 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.4 Production support, acceptance procedures and automation
	ALC_CMS.4 Problem tracking CM coverage
	ALC_DEL.1 Delivery Procedures
	ALC_DVS.1 Identification of security measures
	ALC_FLR.3 Systematic Flaw Remediation
	ALC_LCD.1 Developer defined life-cycle model
	ALC_TAT.1 Well-defined development tools
Class ADV: Development	ADV_ARC.1 Security Architecture Description
	ADV_FSP.4 Complete functional specification
	ADV_IMP.1 Implementation representation of the TSF
	ADV_TDS.3 Basic modular design

Assurance Requirements	
Class AGD: Guidance documents	AGD_OPE.1 Operational user guidance
	AGD_PRE.1 Preparative procedures
Class ATE: Tests	ATE_COV.2 Analysis of coverage
	ATE_DPT.2 Testing: security enforcing modules
	ATE_FUN.1 Functional testing
	ATE_IND.2 Independent testing – sample
Class AVA: Vulnerability assessment	AVA_VAN.3 Focused Vulnerability analysis

7 TOE Summary Specification

This section presents information to detail how the TOE meets the functional and assurance 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 14 - Mapping of TOE Security Functions to Security Functional Requirements

TOE Security Function	SFR ID	Description
Security Audit	FAU_GEN.1	Audit Data Generation
	FAU_GEN.2	User identity association
	FAU_SAR.1	Audit review
	FAU_SAR.2	Restricted audit review
	FAU_SAR.3	Selectable audit review
	FAU_SEL.1	Selective audit
	FAU_STG.1	Protected audit trail storage
	FAU_STG.3	Action in case of possible audit data loss
	FAU_STG.4	Prevention of audit data loss
User Data Protection	FDP_ACC.1	Subset access control
	FDP_ACF.1	Security attribute based access control
	FDP_IFC.1	Subset information flow control
	FDP_IFF.2	Hierarchical security attributes

TOE Security Function	SFR ID	Description
	FDP_RIP.1	Subset residual information protection
Identification and Authentication	FIA_AFL.1	Authentication failure handling
	FIA_ATD.1	User attribute definition
	FIA_SOS.1	Verification of secrets
	FIA_UAU.1	Timing of authentication
	FIA_UAU.7	Protected authentication feedback
	FIA_UID.1	Timing of identification
	FIA_USB.1	User-subject binding
Security Management	FMT_MOF.1(a)	Management of security functions behaviour
	FMT_MOF.1(b)	Management of security functions behaviour
	FMT_MSA.1(a)	Management of security attributes
	FMT_MSA.1(b)	Management of security attributes
	FMT_MSA_(EXT).3	Static attribute initialisation
	FMT_MSA.3	Static attribute initialisation
	FMT_MTD.1(a)	Management of TSF data
	FMT_MTD.1(b)	Management of TSF data
	FMT_REV.1(a)	Revocation
	FMT_REV.1(b)	Revocation
	FMT_SMF.1	Specification of management

TOE Security Function	SFR ID	Description
		functions
	FMT_SMR.1	Security roles
Protection of TOE Security Functions	FPT_STM.1	Reliable time stamps
	FPT_TRC_(EXT).1	Internal TSF consistency
Resource Utilization	FRU_RSA.1	Maximum quotas
IT Environment	FIT_PPC_(EXT).1	IT Environment
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 captures audit data in one of three types of audit files, depending on the type of event:

- Linux OS log files;
- process-specific log files stored on the Host;
- an audit database stored on the SPUs.

7.1.1.1 Linux OS Log Files

There are several Linux OS log files that contain audit data generated by the OS, such as changes to the system time. These log files can only be accessed by a Linux administrator.

7.1.1.2 Process-specific Log Files

All major software components that run on the Host have an associated log. Log files have the following characteristics:

- Each log consists of a set of files stored in a component-specific directory. A separate directory for log files is kept for each process that creates audit logs. Some processes in the TOE are run on a "per session" basis. These subsystems store individual log files on a per session basis with a naming scheme that uniquely identifies which session is being logged.
- Each file contains one day of entries, for a default maximum of seven days.
- Each file contains entries that, at a minimum, have a timestamp, an entry severity type, and a message.

- If an event is related to a specific user or session, that information is stored with the log.

All logs have specified rules on how long each log file is to be retained by the system. The following security relevant audit logs are kept by the TOE:

Table 15 - Security Relevant NPS Audit Logs

Security Relevant NPS Audit Logs	
Backup and Restore Manager	Logs all operations by the nzbackup and nzrestore commands
Bootserver Manager	Logs startup and shutdown of the system and initialization events of all SPUs on the system
Client Manager	Logs all connection requests to the TOE
Database Operation System	Logs all events related to SQL plans submitted to the system
Event Manager	Logs all system level events between the Host and the SPUs
Host Statistics Generator	Logs the starting and stopping of the statistics generator process
Postgres	This is the main database log file. It records information about all database level activities
Startup Server	This log records the startup of all NPS processes and any errors encountered

The logs may be read by an authorized user with appropriate privileges on the Linux OS where the records are stored. The Linux OS also protects the logs from unauthorized access and modification. If the available space for audit storage exceeds a pre-configured level, an alarm is sent to the designated administrators. If the audit trail becomes full, no further audits will be recorded.

7.1.1.3 Audit Database

The audit database tracks information about user interactions with the TOE data, and can be used to generate statistics about users, actions, and data. Information collected in the audit database includes:

- Successful authentication and session creation
- Failed authentication
- Account lock-out action
- All SQL operations
- All administrative operations

Audit database entries log the identity of the process that generated the information, the date and time of the action, the location of the action, the entire command or SQL statement, objects accessed by the commands, the result of the command, and performance information.

Administrators may review the logs, performing searches and sorts based on the values of the audit data fields. Administrators may also determine which events are to be audited, based on the user identity, group identity, event type, object identity, and outcome of those events.

The audit database is stored on the SPUs in tables that use row-level security. Each row is given a label derived from combining two labels: the label of the principal performing the action, and the audit categories associated with that principal. The label of the principal provides protection for the audit data by restricting viewing of the audit to only those users who are permitted to see the original data. The audit categories prevent the principal from viewing the audit data, and allows the audit data to be partitioned among auditors.

The audit database is populated by an audit capture server process. The audit capture server process receives and buffers in memory audit data from other Host processes, and then periodically flushes the data from memory to disk. If the audit disk files used by the audit capture server process become full, the audit capture server process will be unable to write more logs to the files, and will return errors to the processes that are generating the log data. All further activity that requires audit logging will fail until disk space for audit logging has been freed.

The TOE protects the data from tampering by sequencing the data as it is generated by the Host processes. The audit capture server process then further sequences the data as it writes it to disk. In addition, the audit trail is protected by the access control rules in place on the TOE, preventing unauthorized modifications to the audit trail.

TOE Security Functional Requirements Satisfied: FAU_GEN.1, FAU_GEN.2, FAU_SAR.1, FAU_SAR.2, FAU_SAR.3, FAU_SEL.1, FAU_STG.1, FAU_STG.3, FAU_STG.4

7.1.2 User Data Protection

For the purpose of this evaluation, user data is defined as database records stored in all the SPUs. Administratively, the TOE presents its implementation of Discretionary Access Control through the use of an Access Control Matrix (ACM). For all objects in the database, this ACM allows the following access privileges to be assigned: abort, alter, delete, drop, gen stats, list insert, select, truncate, update. These objects may be individual databases, or individual tables[24] within a given database.

The TOE also maintains permissions in the Access Control Matrix that apply globally. These allow permissions to be granted to users or groups that do not relate to specific tables or databases. The privileges that can be granted with this mechanism are: backup, create table, create external table, create group, create materialized group, create sequence, create table, create user, create view, hardware, restore, reclaim, system.

On any operation in the database, the default action is to deny access unless access has been explicitly granted by an authorized administrator. Whenever a subject requests to perform an operation on an object, the ACM is checked to see if the appropriate privilege has been granted. If the privilege has been granted to either the individual or a group of which the individual is a member, then the subject is allowed to perform the operation on the object. If the privilege has not been granted than the request to perform the operation will be denied.

In addition, the TOE implements row-level security through its Label-based Access Control Security Functional Policy. Administrators determine which user data requires row-level security, and configures the SFP to enforce its rules upon that data.

The NPS system supports the concept of a group. A group is categorized as a collection of access rights that have been assigned by an administrator. Individual users can then be given membership in one or more groups. Users who are members of a group inherit all access rights that have been assigned to that group. There is no limit as to the number of groups that can be created or the number of groups that an individual user can be a member of. However, all users are at minimum a member of the group named “Public”.

All user data stored by the TOE exists as a database Object. This can take several forms, for example, a Database, Table, or data contained within one of those objects. All access to this data is mediated by the TOE and subject to access permissions as described above. No direct access to memory or disk storage is provided to end users of TOE.

[24] These objects may also be table-like objects (e.g.: views)

The TOE also provides residual information protection upon allocation of resources to schema objects and non-schema objects that are stored in the system schema.

TOE Security Functional Requirements Satisfied: FDP_ACC.1, FDP_ACF.1, FDP_IFC.1, FDP_IFF.2, FDP_RIP.1

7.1.3 Identification and Authentication

There are two identification and authentication mechanisms used by the TOE. A user may be required to authenticate to the Linux OS in order to perform certain administrative functions. In order to perform queries on the TOE database data, the user must authenticate to the SMP Host Application.

7.1.3.1 Linux Identification and Authentication

System administration is performed using a combination of Linux and NzCLI commands. In order to perform all administrative functions, an authorized administrator must be able to identify and authenticate to the Linux OS as well as the NzCLI.

7.1.3.2 SMP Host Application Identification and Authentication

The TOE performs identification and authentication over each interface to the TOE. No system services (except user login) are available to a user prior to identification and authentication. A user can request services through the nzAdmin or nzCLI interface, directly or via applications enable with the Netezza ODBC, JDBC, or OLE-DB API[25]. Over each of these interfaces the user is required to provide a username and password prior to gaining access to system services. When a user enters login information, the TOE obscures the password feedback so an unauthorized user cannot see the password as it is being entered. All passwords must meet reuse, lifetime, and content metrics as defined by authorized administrators.

Once the user submits the credentials, there are only two possible results: acceptance of a correct set of username and password, or a rejection. It is possible for the TOE to lock access to a user's account if the number of incorrect authentication attempts meets a predefined number set by the Administrator.

A user's identity is bound to one or more groups. This binding is used to determine which privileges this user has been granted. Users may also be granted privileges individually. All decisions on granting access to objects within the TOE are handled by the mechanisms as described in User Data Protection.

The user's credentials can be verified either locally on the Host, or by an LDAP server connected to the Host via an LDAPS connection. When using LDAP to authenticate users, the TOE passes the user's credentials (username and password) to the LDAP server. The LDAP server verifies the credentials against the directory, and returns a "success" or "fail" indicator to the TOE. If the authentication is successful, the TOE proceeds to grant the appropriate privileges, as described above.

TOE Security Functional Requirements Satisfied: FIA_AFL.1, FIA_ATD.1, FIA_SOS.1, FIA_UAU.1, FIA_UAU.7, FIA_UID.1 FIA_USB.1

7.1.4 IT Environment

The TOE operating system is compliant with the requirements of the Controlled Access Protection Profile, version 1.d (CAPP), in addition to applicable requirements specified in this Security Target. The security functions for this PP include requirements covering auditing, user data protection, identification and authentication, security

[25] API – Application Programming Interface

management, and protection of the TOE security functions. The requirements claimed in this ST are in addition to those claimed by the CAPP.

TOE Security Functional Requirements Satisfied: FIT_PPC_(EXT).1,

7.1.5 Security Management

This section discusses the TOE's role definition and role management functionalities. Strictly speaking, there are only two "roles" enforced by the TOE. These are the "ADMIN account" and other "Administrator defined groups". The "ADMIN account" is a special account that posses all rights and privileges available to the system. The other "Administrator defined groups" can be divided into two logical roles: administrators and database users. Administrators are users that have at least some administrative privileges. Database users have no administrative privileges. When the TOE is first installed, the installer is logged in as 'ADMIN'. This role has full access to all functionality in the TOE, and should only be used during initial configuration to create other administrative accounts. The other administrative users can then finish the TOE configuration and create the appropriate database users and privileges.

All access rights within the TOE are granted based upon the User Data Protection mechanisms provided through the access control security functional policies. The privileges that can be assigned through these policies are described in more detail in section 7.1.2.

The 'ADMIN', or another user granted appropriate privileges, can perform all administrative activities necessary to manage the TOE. By using an ACM instead of predefined roles, it is easier to maintain the concept of least privilege. Each user is only given the exact rights they need at that time and if an Administrator needs to assign rights to a large number of users, they can still create a group, and assign the rights to the group. This allows administrators to customize groups to their specific needs.

TOE Security Functional Requirements Satisfied: FMT_MOF.1(a), FMT_MOF.1(b), FMT_MSA.1(a), FMT_MSA.1(b), FMT_MSA.3, FMT_MSA_(EXT).3, FMT_MTD.1(a), FMT_MTD.1(b), FMT_REV.1(a), FMT_REV.1(b), FMT_SMF.1, FMT_SMR.1.

7.1.6 Protection of the TSF

The TOE provides several mechanisms for protecting its security functions. The system has redundancies in case of a hardware failure, and to protect data stored on the SPUs.

The TOE provides reliable timestamp information for its own use. The time is set through the use of an NTP client, or manually to the Linux OS. From there, other subsystems are able to retrieve the time for inclusion in audit records.

TOE Security Functional Requirements Satisfied: FPT_STM.1, FPT_TRC_(EXT).1.

7.1.7 Resource Utilization

The TOE enforces maximum quotas on the duration of a user's session, the duration of a query, the number of rows that can be returned from a single query, and the percentage of contended system resources that a class of users can use.

TOE Security Functional Requirements Satisfied: FRU_RSA.1.

7.1.8 TOE Access

The TOE allows only an administrator-configurable maximum number of concurrent TOE sessions for any user. In addition, TOE administrators may define when a user may be denied access to the TOE, based on user identity or group identity, time of day, and day of week. The TOE will also store and retrieve the date and time of the last successful and unsuccessful attempts to access the TOE since the last successful session establishment by each user.

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 Parts 2 and 3 of the Common Criteria Standard for Information Technology Security Evaluations, version 3.1, revision 2. There are four extended SFRs contained within this ST: FIT_PPC_(EXT).1, FMT_MSA_(EXT).3, FPT_TRC_(EXT).1, and FTA_TAH_(EXT).1. These were included to define the Operating System PP compliance, management activities, internal TSF consistency, and TOE access history by the TOE. Although the PP lists an additional four extended SFRS: FAU_GEN.1-NIAP-0410, FAU_SEL.1-NIAP-0407, FAU_SEL.1-NIAP-0407, and FAU_ACF.1-NIAP-0407, the names were modified in this ST to match the standard CC v3.1 SFR names. The wording of the PP SFRs was consistent with the wording of the standard SFRs, so using the extended SFR names was not necessary.

This Security Target claims demonstrable conformance with U.S. Government Protection Profile for Database Management Systems in Basic Robustness Environments, Version 1.2. The following describes the detailed rationale for demonstrable conformance for each relevant section of the ST:

Security Problem Definition: All of the Threats, Organizational Security Policies, and Assumptions identified in the PP are included in this ST and are in accordance with the requirements of the PP. Some additional Threats, Organizational Security Policies, and Assumptions have been included in the ST that were not included in the PP. The assumption A.OS_PP_VALIDATED was added because the OS is part of the TOE.

Security Objectives: All the TOE Security Objectives and Environment Security Objectives identified in the PP are included in this ST and are in accordance with the requirements of the PP. Some additional Security Objectives for the TOE, IT Security Objectives, and Non-IT Security Objectives have been included in the ST that were not included in the PP. The Environment Security Objective OE.OS_PP_VALIDATED was changed to Security Objective O.OS_PP_VALIDATED because the OS is part of the TOE.

Security Functional Requirements: All the SFRs identified in the PP are included in this ST (some with name changes or notes), and all the operations applied to the SFRs derived from the PP are in accordance with the requirements of the PP. Some additional SFRs have been included in the ST that were not included in the PP.

Security Assurance Requirements: All the Security Assurance Requirements identified in the PP are included in this ST and are in accordance with the requirements of the PP. Some additional Security Assurance Requirements have been included in the ST that were not included in the PP.

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, polices, and assumptions to the security objectives is 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 16 - Threats: Objectives Mapping

Threats	Objectives	Rationale

Threats	Objectives	Rationale
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.MASQUERADE A user or process may masquerade as another entity in order to gain unauthorized access to data or TOE resources.	O.TOE_ACCESS The TOE will provide mechanisms that control a user's logical access to the TOE.	The objective 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 mechanism, this objective helps mitigate the possibility of a user attempting to login and masquerade as an authorized user. In addition, this 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.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.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.
	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	O.CONFIGURATION_IDENTIFICATI	O.CONFIGURATION_IDENTIFICATION plays a role in countering this

Threats	Objectives	Rationale
Unintentional errors in implementation of the TOE design may occur, leading to flaws that may be exploited by a casually mischievous user or program.	ON 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.	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 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

Threats	Objectives	Rationale
		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 in a protected resource within its Scope of Control is not released when the resource is allocated.	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.
T.TSF_COMPROMISE A user or process may cause, through an unsophisticated attack, TSF data, or executable code 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 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 administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.	O.MANAGE is necessary because an access control policy 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_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.PARTIAL_SELF_PROTECTION ensures the TOE is capable of protecting itself from attack.
	O.RESIDUAL_INFORMATION The TOE will ensure that any information contained in a protected resource within its Scope of Control is	O.RESIDUAL_INFORMATION is necessary to mitigate this threat, because even if the security mechanisms do not allow a user to view TSF data, if TSF data were to

Threats	Objectives	Rationale
	not released when the resource is allocated.	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 are not authorized according to the TOE security policy.	O.ACCESS_HISTORY The TOE will store and retrieve information (to authorized users) related to previous attempts to establish a session.	O.ACCESS_HISTORY is important to mitigate this threat because it ensures the TOE will be able to store and retrieve the information that will advise the user of the last successful login attempt and actions performed without their knowledge.
	O.MEDIATE The TOE must protect user data in accordance with its security policy.	O.MEDIATE ensures that all accesses to user data are subject to mediation, unless said data has been specifically identified as public data. The TOE requires successful authentication to the TOE prior to gaining access to any controlled-access content. By implementing strong authentication to gain access to these services, an attacker's opportunity to conduct a man-in-the-middle or password guessing attack successfully is greatly reduced. 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 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 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 administrators in their management of the security of the	The threat of an authorized administrator failing to know about malicious audit events produces the objectives of the authorized administrator having the capability to use the mechanisms (O.MANAGE) to

Threats	Objectives	Rationale
	TOE, and restrict these functions and facilities from unauthorized use.	review audit records.
T.RESOURCE An authenticated database user may consume global database resources, in a way that compromises the ability of other database users to access the DBMS.	O.RESOURCE The TOE must provide the means of controlling the consumption of database resources by authorized users of the TOE.	The objective O.RESOURCE ensures that individual users cannot use more of specific resources than defined in their quota. An authorized administrator that can assign quotas to users can use this function to ensure that a sufficient amount of resources of a specific kind is always available, allowing authorized users to use the DBMS at any time they are allowed by the TOE policy to use it.
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_PROTECTION The TOE will provide the capability to protect audit information.	The objective O.AUDIT_PROTECTION helps to mitigate the threat T.AUDIT_COMPROMISE by protecting the audit trail from unauthorized access and loss of audit records.
T.LBAC An authorized database user may access labeled information contained within a database without having the authorization to access that information.	O.ACCESS_LBAC The TOE must provide the ability for labels to be associated with subjects and database objects in accordance with the P.LABEL security policy. For entities that have been associated with labels, the TOE must use these labels as a basis for implementing an information flow control policy in accordance with the P.INFOFLOW policy.	The objective O.ACCESS_LBAC helps to mitigate the threat T.LBAC by ensuring that labels are associated with subjects and database objects in accordance with the P.LABEL security policy, and must therefore be subject to the P.INFOFLOW policy.
T.POOR_DEVELOPMENT_ENVIRONMENT The TOE's development environment may not protect the TOE and its parts during development and maintenance, may not ensure the TOE meets its SFRs, and may implement ill-defined, inconsistent, or incorrect development tools to develop the TOE.	O.DEVELOPMENT_ENVIRONMENT The TOE's development environment will protect the TOE and its parts during development and maintenance, ensure that the TOE meets its SFRs, and prevent ill-defined, inconsistent, or incorrect development tools from being used to develop the TOE.	O.DEVELOPMENT_ENVIRONMENT ensures that the TOE's development environment protects the TOE from improper development and maintenance by requiring that well-documented tools and techniques are employed. In addition, a life-cycle model is employed, and the environment is secure physically and procedurally. Finally, only authorized employees are able to access the TOE and its parts in the development environment.

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 17 - 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.
	O.AUDIT_REVIEW The TOE must provide the means of reviewing the audit log entries allowing users with the required access rights to the audit log to evaluate the audit log entries.	O.AUDIT_REVIEW helps to address this policy by providing authorized administrators with the ability to selectively review the audit log information.
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 isolate 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).
P.LABEL Labels can be associated with subjects and with storage objects which are rows within tables: a) A label is composed of a hierarchical level (classification), a	O.ACCESS_LBAC The TOE must provide the ability for labels to be associated with subjects and database objects in accordance with the P.LABEL security policy. For entities that have been associated with labels, the TOE must use these	O.ACCESS_LBAC addresses this policy by providing the ability for labels to be associated with subjects and database objects according to the P.LABEL policy.

Policies	Objectives	Rationale
set of non-hierarchic categories, and a set of hierarchic groups, as determined by the organization that owns the information stored in the database. b) A storage object label reflects the sensitivity of the information stored in the object. C) A subject label reflects the authorization of the subject to access the organization's labeled information according to defined access rules.	labels as a basis for implementing an information flow control policy in accordance with the P.INFOFLOW policy.	
	OE.USERS Those responsible for the TOE must ensure that users are assigned label authorizations and policy privileges commensurate with the degree of trust placed in them by the organization that owns, or is responsible for, the information processed by or stored in the TOE.	OE.USERS supports this OSP by ensuring that administrators assign appropriate label authorisations and policy privileges to users in accordance with P.LABEL.
P.INFOFLOW Information flow from entity A to entity B shall be permitted only if it does not result in a subject being able to observe labeled information that the subject is not authorized to see.	O.ACCESS_LBAC The TOE must provide the ability for labels to be associated with subjects and database objects in accordance with the P.LABEL security policy. For entities that have been associated with labels, the TOE must use these labels as a basis for implementing an information flow control policy in accordance with the P.INFOFLOW policy.	O.ACCESS_LBAC addresses this policy by providing the ability for labels to be associated with subjects and database objects, and requiring that the TOE use these labels as a basis for implementing an information flow control policy in accordance with P.INFOFLOW. This prevents subjects from seeing labeled information that the subject is not authorized to see.
	OE.USERS Those responsible for the TOE must ensure that users are assigned label authorizations and policy privileges commensurate with the degree of trust placed in them by the organization that owns, or is responsible for, the information processed by or stored in the TOE.	OE.USERS supports the OSP by ensuring that administrators assign appropriate label authorisations and policy privileges to users.
P.OS_PP_VALIDATED The underlying OS has been validated against an NSA-sponsored OS PP of at least Basic Robustness.	O.OS_PP_VALIDATED The underlying OS has been validated against an NSA -sponsored OS PP of at least Basic Robustness.	The TOE's OS must be validated to at least basic robustness to ensure it provides an appropriate level of protection for the DBMS. The OS must provide domain separation, non-bypassability, audit review, audit storage, and identification and authentication.

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.

P.OS_PP_VALIDATED was changed from an assumption (as listed in the PP) to a policy because O.OS_PP_VALIDATED cannot map to an assumption. O.OS_PP_VALIDATED was changed from an environmental objective (as listed in the PP) because the operating system is part of the TOE, not the TOE environment.

8.2.3 Security Objectives Rationale Relating to Assumptions

Table 18 - Assumptions: Objectives Mapping

Assumptions	Objectives	Rationale
A.NO_EVIL Sites using the TOE shall ensure that authorized administrators are non-hostile, appropriately trained, and follow all administrator guidance.	OE.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 investigations 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 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.	OE.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 individual that is authorized to access the TOE environment.
A.MIDTIER To ensure accountability in multi-tier environments, any middle-tiers will pass the original client ID through to the TOE.	OE.NO_EVIL Sites using the TOE shall ensure that authorized administrators are non-hostile, appropriately trained, and follow all administrator guidance.	OE.NO_EVIL requires that any middle-tier must pass the original client ID through to the TOE, and advises the administrator how to configure this functionality correctly.
A.DIR_PROT The directory server used by the	OE.DIR_CONTROL The directory server must provide	OE.DIR_CONTROL ensures that the directory server used to store information for users is protected from

Assumptions	Objectives	Rationale
TOE provides protection mechanisms against unauthorized access to TSF data stored in the directory. This includes the assumptions that queries are properly authenticated, that the TSF data stored in the directory is protected by the access control mechanisms of the directory server, that the TSF data in the directory server is properly managed by the administrative personnel, and that the directory server as well as its network connections are physically and logically protected from access and interference by unauthorized persons.	access control mechanisms to prohibit unauthorized access to directory entries. The directory server must authenticate users before it allows them to access TSF data stored in the directory.	unauthorized access and managed correctly, thereby upholding this assumption.
A.DIR_MGMT The information about users stored in the directory (password verifier, password policy, and privileges) is managed correctly by authorized personnel.	OE.DIR_CONTROL The directory server must provide access control mechanisms to prohibit unauthorized access to directory entries. The directory server must authenticate users before it allows them to access TSF data stored in the directory.	OE.DIR_CONTROL allows administrators to restrict access to the information on users stored in the directory to defined users, thereby upholding this assumption.
A.COM_PROT Internal TSF communication as well as communication between the TOE and the directory server are protected from unauthorized access to the transmitted data and ensure that the communication peers are the intended ones.	OE.COM_PROT The environment must provide protection mechanisms that prohibit unauthorized access to data the TOE transfers over communication links. This applies to data the TOE transmits to another part of itself as well as data exchanged between the TOE and the external directory server. This protection may be provided by physical protection, logical protection, or a combination of both.	OE.COM_PROT ensures that communication links between distributed parts of the TOE, as well as communication links between the TOE and the external directory server are protected, thereby upholding this assumption.

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 Refinements of Security Functional Requirements

The following refinements of Security Functional Requirements from CC version 3.1 have been made to clarify the content of the SFRs, and make them easier to read:

FAU_GEN.2.1: The terms “and/or identified groups” and “and/or group” have been added to the SFR to include audited events that are identified by Group ID rather than user id.

FDP_ACF.1.1: The terms “subject privileges” and “object access privileges” have been added for clarity of security attributes referred to in FMT_REV.1(b).

FDP_IFF.2.6: The term “security attribute” was refined to “security labels” to better identify the attributes in question.

FIA_ATD.1.1: The phrase “Permissions granted to the user as defined by” has been added to clarify the actual capabilities of the TOE with regard to this SFR. This wording enhances the understanding of the TOE functionality, while retaining the spirit of the SFR as intended by the Protection Profile.

FMT_MSA.3.1: The term “database object” was added to the SFR to better identify the security attributes in question.

FMT_MSA.3.2: The term “for Label-based Access Control security attributes” was added to better identify the values in question; the term “an object or information” was refined to “a database object” to better identify the object in question.

8.4 Rationale for Extended Security Functional Requirements

Four extended TOE SFRs were created to specifically address functionality that is not fully represented by the standard Common Criteria requirements.

8.4.1 Rationale for TOE Extended Security Functional Requirements

FIT_PPC_(EXT).1 is necessary to ensure the TOE will be running on an OS that is at least as robust as the TOE itself.

The CC does not allow the ST author to specify restrictive values that are not modifiable. FMT_MSA_(EXT).3 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.

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 this ST.

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 this ST’s TOE, this in 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 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 this ST’s environment in that it is not the intent of the authors to dictate that distributed TSF components keep track of connected or disconnected components.

In general, to meet the needs of this ST, it is acceptable to only require a mechanism that provides TSF data consistency in a timely manner after it is determined that it is consistent.

This ST does not require the TOE to contain a client. Therefore, the ST cannot require the client to display a message. FTA_TAH_(EXT).1 has been modified to require the TOE to store and retrieve the access history instead of displaying it.

8.5 Rationale for Extended TOE Security Assurance Requirements

No extended Security Assurance Requirements have been defined for this Security Target.

8.6 Security Requirements Rationale

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

8.6.1 Rationale for Security Functional Requirements of the TOE Objectives

Table 19 - Objectives:SFRs Mapping

Objective	Requirements Addressing the Objective	Rationale
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 his 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 user is attempting to access her account. These records should not be deleted until after the user has been notified of his access history.
O.ADMIN_ROLE The TOE will provide authorized administrator roles to isolate administrative actions.	FMT_SMR.1 Security roles	The TOE will establish, at least, an authorized administrator role. 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.
O.AUDIT_GENERATION The TOE will provide the capability to detect and create records of security relevant events associated with users.	FAU_GEN.1 Audit Data Generation	FAU_GEN.1 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 in the TOE. This requirement also defines the

Objective	Requirements Addressing the Objective	Rationale
		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 the additional security functional requirements the ST author has added to the PP.
	FAU_GEN.2 User identity association	FAU_GEN.2 ensures that the audit records associate a user 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.
	FAU_SEL.1 Selective audit	FAU_SEL.1 allows the administrator to configure which auditable events will be recorded in the audit trail. This provides the administrator with flexibility in recording only those events that are deemed necessary by site policy, thus reducing the number of resources consumed by the audit mechanism.
	FMT_MTD.1(a) Management of TSF data	FMT_MTD.1(a) ensures that authorized administrators have the ability to include or exclude auditable events from the audit trail.
	FPT_STM.1 Reliable time stamps	FPT_STM.1 ensures that a reliable date and time is available for use in the audit records.
O.MANAGE The TOE will provide all the functions and facilities necessary to support the authorized administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.	FMT_MOF.1(a) Management of security functions behaviour	FMT_MOF.1(a) requires that the ability to use particular TOE capabilities be restricted to the administrator.
	FMT_MSA.1(a) Management of security attributes	FMT_MSA.1(a) 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.

Objective	Requirements Addressing the Objective	Rationale
	FMT_MTD.1(a) Management of TSF data	FMT_MTD.1(a) requires that the ability to manipulate TOE content is restricted to administrators.
	FMT_MTD.1(b) Management of TSF data	FMT_MTD.1(b) requires that the ability to manipulate TOE content is restricted to administrators.
	FMT_REV.1(a) Revocation	FMT_REV.1(a) restricts the ability to revoke attributes to the administrator.
	FMT_REV.1(b) Revocation	FMT_REV.1(b) restricts the ability to revoke attributes to the administrator and authorized database users.
	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.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, 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 subject and object covered are defined by the TOE's policy.
	FDP_ACF.1 Security attribute based access control	FDP_ACF.1 defines the security attribute used to provide access control to objects based on the TOE's access control 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

Objective	Requirements Addressing the Objective	Rationale
		requirement is to maintain consistency of replicated TSF data.
O.RESIDUAL_INFORMATION The TOE will ensure that any information contained in a protected resource within its Scope of Control is not released when the resource is allocated.	FDP_RIP.1 Subset residual information protection	FDP_RIP.1 is used to ensure the contents of resources are not available to subjects other than those explicitly granted access to the data.
O.TOE_ACCESS The TOE will provide mechanisms that control a user's logical access to the TOE.	FIA_AFL.1 Authentication failure handling	FIA_AFL.1 ensures that a user cannot keep entering an invalid password in attempts to login; this will prevent a brute force attack to crack a user's password.
	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.
	FIA_SOS.1 Verification of secrets	FIA_SOS.1 ensures that passwords for local users meet defined metrics.
	FIA_UAU.1 Timing of authentication	FIA_UAU.1 requires that all users must authenticate before they are given access to the TOE.
	FIA_UAU.7 Protected authentication feedback	FIA_UAU.7 ensures that only obscured feedback is given to the user while attempting to authenticate to the TOE.
	FIA_UID.1 Timing of identification	FIA_UID.1 requires that users must uniquely identify themselves before they are given access to the TOE.
	FIA_USB.1 User-subject binding	FIA_USB.1 ensures that the TOE will bind users' security attributes with the users, and enforce rules on the initial association of those security attributes to the users, and changes to those security attributes.

Objective	Requirements Addressing the Objective	Rationale
	FMT_MTD.1(b) Management of TSF data	FMT_MTD.1(b) ensures that only authorized administrators have the ability to manage the TSF data.
	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.
	FTA_TSE.1 TOE session establishment	FTA_TSE.1 allows the TOE to restrict access to the TOE based on certain criteria.
O.RESOURCE The TOE must provide the means of controlling the consumption of database resources by authorized users of the TOE.	FRU_RSA.1 Maximum quotas	FRU_RSA.1 ensures that the TSF enforces maximum quotas for specified resources on individual users.
O.AUDIT_REVIEW The TOE must provide the means of reviewing the audit log entries allowing users with the required access rights to the audit log to evaluate the audit log entries.	FAU_SAR.1 Audit review	FAU_SAR.1 ensures that users with read access to the audit records are able to review audit records for which they have access.
	FAU_SAR.2 Restricted audit review	FAU_SAR.2 ensures that the audit trail is protected so that only authorized users may access it.
	FAU_SAR.3 Selectable audit review	FAU_SAR.3 ensures that the TSF provides the capability to audit the actions of an individual user, and allows administrators to review those actions.
	FPT_STM.1 Reliable time stamps	FPT_STM.1 ensures that the time stamp associated with the audit records is reliable.
O.AUDIT_PROTECTION The TOE will provide the capability to protect audit information.	FAU_STG.1 Protected audit trail storage	FAU_STG.1 ensures that the stored audit records are protected from unauthorized deletion or modification, thereby allowing administrators to review the audit logs.
	FAU_STG.3 Action in case of possible audit data	FAU_STG.3 ensures that authorized administrators are alerted when the audit trail reaches a configurable limit

Objective	Requirements Addressing the Objective	Rationale
	loss	on size, thereby allowing the administrators to take action to protect the audit trail.
	FAU_STG.4 Prevention of audit data loss	FAU_STG.4 ensures that audited events will no longer be generated when the audit trail is full, allowing administrators to view all audit records before they are overwritten.
O.ACCESS_LBAC The TOE must provide the ability for labels to be associated with subjects and database objects in accordance with the P.LABEL security policy. For entities that have been associated with labels, the TOE must use these labels as a basis for implementing an information flow control policy in accordance with the P.INFOFLOW policy.	FDP_IFC.1 Subset information flow control	FDP_IFC.1 ensures that the Label-based Access Control SFP attributes and rules have a defined scope of control.
	FDP_IFF.2 Hierarchical security attributes	FDP_IFF.2 ensures that the attributes and rules for the Label-based Access Control SFP are defined.
	FMT_MOF.1(b) Management of security functions behaviour	FMT_MOF.1(b) ensures that only authorized administrators have the ability to modify the behaviour of the Label-based Access Control SFP functions.
	FMT_MSA.1(b) Management of security attributes	FMT_MSA.1(b) ensures that only suitably privileged users have the ability to modify the labels and privileges enforced by the Label-based Access Control SFP.
	FMT_MSA.3 Static attribute initialisation	FMT_MSA.3 ensures that no default values for database object security attributes are defined for the Label-based Access Control SFP, and that no database users are permitted to specify alternative initial values for Label-based Access Control SFP security attributes when a database object is created.
O.OS_PP_VALIDATED The underlying OS has been validated against an NSA - sponsored OS PP of at least Basic Robustness.	FIT_PPC_(EXT).1 IT Environment	FIT_PPC_(EXT).1 states the TOE's OS must be validated against an OS PP of at least basic robustness.

8.6.2 Security Assurance Requirements Rationale

EAL4+ was selected because it is best suited to addressing the stated security objectives. EAL4+ challenges vendors to use best (rather than average) commercial practices. EAL4+ allows the vendor to evaluate the product at a detailed level while avoiding the non-trivial expense and rigor of higher assurance levels and still benefitting from the Common Criteria Recognition Agreement. The chosen assurance level is appropriate for the threats defined in the environment.

The augmentation of ALC_FLR.3 was chosen to give greater assurance of the developer's on-going flaw remediation processes.

Table 20 - Objectives: SARs Mapping

Objective	Requirements Addressing the Objective	Rationale
O.ADMIN_GUIDANCE The TOE will provide administrators with the necessary information for secure management.	ALC_DEL.1 Delivery procedures	ALC_DEL.1 ensures that the administrator is provided documentation that instructs him how to ensure that 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 her 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.
	AGD_PRE.1 Preparative procedures	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 guidance the result is a TOE in a secure configuration.
	AGD_OPE.1 Operational user guidance	AGD_OPE.1 mandates the developer provide the administrator with guidance on how to operate the TOE in a 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

Objective	Requirements Addressing the Objective	Rationale
		description of how to set up 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 or alerts, and act accordingly.
	AGD_OPE.1 Operational user guidance	AGD_OPE.1 is also intended for non-administrative users, so 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 Operational user guidance AGD_PRE.1 Preparative procedures	AGD_OPE.1 and AGD_PRE.1 analysis during evaluation will ensure that the guidance documentation is complete and consistent, and notes all requirements for external security measures.
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.	ALC_CMC.4 Product support, acceptance procedures and automation ALC_CMS.4 Problem tracking CM coverage	ALC_CMC.4 and ALC_CMS.4 address this objective by requiring that there be a unique reference for the TOE, and that the TOE is labeled with that reference. They also require that there be an automated 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.3 Systematic flaw remediation	ALC_FLR.3 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 and to registered users of the system.
O.DOCUMENTED_DESIGN The design of the TOE is adequately and accurately documented.	ADV_FSP.4 Complete functional specification	ADV_FSP.4 requires that the interfaces to the TOE be documented and specified.
	ADV_TDS.3 Basic modular design	ADV_TDS.3 requires the high-level and low-level design of the TOE be documented and specified, and that said design be shown to correspond to the interfaces. In addition, there

Objective	Requirements Addressing the Objective	Rationale
		must be a correspondence between adjacent layers of the design decomposition.
O.INTERNAL_TOE_DOMAINS The TSF will maintain internal domains for separation of data and queries belonging to concurrent users.	ADV_ARC.1 Security architecture description ADV_IMP.1 Implementation representation of the TSF	ADV_ARC.1 provides the security architecture description of the security domains maintained by the TSF that are consistent with the SFRs. ADV_IMP.1 provides the implementation representation of the TSF. 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_FUNCTIONAL_TEST The TOE will undergo some security functional testing that demonstrates the TSF satisfies some of its security functional requirements.	ATE_COV.2 Analysis of coverage	ATE_COV.2 requires that there be a correspondence between the tests in the test documentation and the TSF subsystems as described in the TOE design.
	ATE_DPT.2 Testing: security enforcing modules	ATE_DPT.2 requires that there be a correspondence between the tests in the test documentation and the TSF modules as described in the TOE design.
	ATE_FUN.1 Functional testing	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. They require that the developer run those tests, and show that the expected results were achieved.
	ATE_IND.2 Independent testing – sample	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.PARTIAL_SELF_PROTECTION	ADV_ARC.1	ADV_ARC.1 provides the security architecture description of the security

Objective	Requirements Addressing the Objective	Rationale
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.	Security architecture description	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.VULNERABILITY_ANALYSIS The TOE will undergo some vulnerability analysis to demonstrate the design and implementation of the TOE does not contain any obvious flaws.	AVA_VAN.3 Focused vulnerability analysis	The AVA_VAN.3 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.3 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 enhanced-basic attack potential to violate the TOE's security policies.
O.DEVELOPMENT_ENVIRONMENT The TOE's development environment will protect the TOE and its parts during development and maintenance, ensure that the TOE meets its SFRs, and prevent ill-defined, inconsistent, or incorrect development tools from being used to develop the TOE.	ALC_DVS.1 Identification of security measures ALC_LCD.1 Developer defined life-cycle model ALC_TAT.1 Well-defined development tools	ALC_DVS.1 requires that physical, procedural, personnel, and other security measures be used in the development environment to protect the TOE and its parts, ensuring the confidentiality and integrity of the TOE design and implementation in its development environment. ALC_LCD.1 requires that the developer establish a life-cycle model for the development and maintenance of the TOE, ensuring that the TOE meets all of its SFRs. ALC_TAT.1 requires that the developer select tools and institute techniques in the development environment are well-defined and appropriate for the development of the TOE.

8.6.3 Dependency Rationale

This ST does satisfy all the requirement dependencies of the Common Criteria. Table 21 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 21 - Functional Requirements Dependencies

SFR ID	Dependencies	Dependency Met	Rationale
FAU_GEN.1	FPT_STM.1	✓	
FAU_GEN.2	FAU_GEN.1	✓	
	FIA_UID.1	✓	
FAU_SAR.1	FAU_GEN.1	✓	
FAU_SAR.2	FAU_SAR.1	✓	
FAU_SAR.3	FAU_SAR.1	✓	
FAU_SEL.1	FAU_GEN.1	✓	
	FMT_MTD.1	✓	FMT_MTD.1(a) and FMT_MTD.1(b) satisfy this dependency.
FAU_STG.1	FAU_GEN.1	✓	
FAU_STG.3	FAU_STG.1	✓	
FAU_STG.4	FAU_STG.1	✓	
FDP_ACC.1	FDP_ACF.1	✓	
FDP_ACF.1	FDP_ACC.1	✓	
	FMT_MSA.3	✓	FMT_MSA_(EXT).3 satisfies this dependency.

SFR ID	Dependencies	Dependency Met	Rationale
FDP_IFC.1	FDP_IFF.1	✓	FDP_IFF.2 is hierarchical to FDP_IFF.1, so the inclusion of FDP_IFF.2 satisfies this dependency.
FDP_IFF.2	FDP_IFC.1	✓	
	FMT_MSA.3	✓	
FDP_RIP.1	No dependencies		
FIA_AFL.1	FIA_UAU.1	✓	
FIA_ATD.1	No dependencies		
FIA_SOS.1	No dependencies		
FIA_UAU.1	FIA_UID.1	✓	
FIA_UAU.7	FIA_UAU.1	✓	
FIA_UID.1	No dependencies		
FIA_USB.1	FIA_ATD.1	✓	
FIT_PPC_(EXT).1	No dependencies		
FMT_MOF.1(a)	FMT_SMF.1	✓	
	FMT_SMR.1	✓	
FMT_MOF.1(b)	FMT_SMR.1	✓	
	FMT_SMF.1	✓	
FMT_MSA.1(a)	FMT_SMF.1	✓	
	FDP_ACC.1	✓	

SFR ID	Dependencies	Dependency Met	Rationale
	FMT_SMR.1	✓	
FMT_MSA.1(b)	FDP_IFC.1	✓	
	FMT_SMF.1	✓	
	FMT_SMR.1	✓	
FMT_MSA_(EXT).3	FMT_MSA.1	✓	This dependency is satisfied by FMT_MSA.1(a).
	FMT_SMR.1	✓	
FMT_MSA.3	FMT_MSA.1	✓	This dependency is satisfied by FMT_MSA.1(b).
	FMT_SMR.1	✓	
FMT_MTD.1(a)	FMT_SMF.1	✓	
	FMT_SMR.1	✓	
FMT_MTD.1(b)	FMT_SMF.1	✓	
	FMT_SMR.1	✓	
FMT_REV.1(a)	FMT_SMR.1	✓	
FMT_REV.1(b)	FMT_SMR.1	✓	
FMT_SMF.1	No dependencies		
FMT_SMR.1	FIA_UID.1	✓	
FPT_STM.1	No dependencies		
FPT_TRC_(EXT).1	No dependencies		

SFR ID	Dependencies	Dependency Met	Rationale
FRU_RSA.1	No dependencies		
FTA_MCS.1	FIA_UID.1	✓	
FTA_TAH_(EXT).1	No dependencies		
FTA_TSE.1	No dependencies		

9 Acronyms and Terminology

9.1.1 Acronyms

Table 22 - Acronyms

Acronym	Definition
ACM	Access Control Matrix
AMPP	Asymmetric Massively Parallel Processing
API	Application Programming Interface
BI	Business Intelligence
CAPP	Controlled Access Protection Profile
CC	Common Criteria
CEM	Common Evaluation Methodology
CLI	Command Line Interface
DAC	Discretionary Access Control
DB	Database
DBA	Database Administrator
DBMS	Database Management System
EAL	Evaluation Assurance Level
GUI	Graphical User Interface
HA	High Availability
ID	Identifier
IT	Information Technology
JDBC	Java Database Connectivity
KVM	Keyboard Video Mouse

Acronym	Definition
LDAP	Lightweight Directory Access Protocol
MLS	Multi-level Security
MPP	Massive Parallel Processing
NIC	Network Interface Card
NPS	Netezza Performance Server
NSA	National Security Agency
NTP	Network Time Protocol
ODBC	Open Database Connectivity
OLE-DB	Object Linking and Embedding Database
OS	Operating System
OSP	Organizational Security Policy
PP	Protection Profile
SAR	Security Assurance Requirement
SFP	Security Functional Policy
SFR	Security Functional Requirement
SMP	Symmetric Multiprocessing
SPU	Snippet Processing Units
SQL	Structured Query Language
ST	Security Target
TOE	Target of Evaluation
TSF	TOE Security Function

9.1.2 Terminology

Common Criteria Policy – The term 'policy' in Common Criteria is used to refer to a Security Functional Policy (SFP), which is the security policy enforced by a particular Security Function (SF).

NPS Policy – The term 'policy' in TOE guidance is used to refer to policies established by a database administrator to specify how Label-based Access Control is to be enforced on a database. Such a policy will always be referred to in this document via the phrase "NPS policy".

Discretionary Access Control – A kind of access control that restricts access to objects based on the identity of the subjects or groups to which they belong, and in which subjects are capable of passing their own permissions on to any other subjects.

Label-based Access Control – A kind of access control that constrains the ability of a subject to access or perform operations on objects.

Schema objects - objects stored in databases. In the TOE, these include tables, views, sequences, user-defined functions, user-defined aggregates, row secure tables, etc.

Non-schema objects - global metadata objects, such as users, groups, categories, cohorts, and labels.

Sys schema - the SYSTEM database where the metadata describing the schema objects and non-schema objects is stored.