Netezza Corporation
Netezza Performance Server version 3.0

Netezza Security Target

Document Version 1.1

Prepared for:

Netezza Corporation
200 Crossing Blvd.
Framingham, MA 01702
Phone: (508) 665-6800
Fax: (508)665-6811

Prepared by:

Corsec Security, Inc.
10340 Democracy Lane, Suite 201
Fairfax, VA 22030
Phone: (703) 267-6050
Fax: (703) 267-6810

Revision History

Version	Modification Date	Modified By	Description of Changes
0.1	2006-01-19	Jonathan Halperin	Initial draft.
0.2	2006-06-05	Matthew Appler	Minor revisions
0.3	2006-09-06	Christie Kummers	Minor revisions. Addressed ORs.
0.4	2006-09-27	Christie Kummers	Removed all footnotes that contained acronyms and placed the acronyms in the Acronym section of the ST.
0.5	2007-05-11	Teresa MacArthur	Addressed verdicts dated 27 March 2007
0.6	2007-06-13	Teresa MacArthur	Addressed verdicts dated 6 June 2007.
0.7	2007-07-03	Amy Nicewick	Added model numbers of evaluated appliances.
1.0	2007-08-20	Amy Nicewick	Removed list of assurance requirements, added trademark to name, changed Part 2 and 3 designations.
1.1	2007-09-04	Amy Nicewick	Updated software version number in Table 1.

Table of Contents

Table of Figures

Table of Tables

1 Security Target Introduction

This section provides the Target of Evaluation (TOE) identification, ST conventions, ST conformance claims, and the ST organization. The Target of Evaluation is the Netezza Performance Server version 3.0, Release 6 [Build 6414], 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 contains the following sections to provide mapping of the Security Environment to the Security Requirements that the TOE meets in order to remove, diminish, or mitigate the defined threats:

- Security Target Introduction (Section 1) – Provides a brief summary of the content of the ST and describes the organization of other sections of this document.
- TOE Description (Section 2) – Provides an overview of the TOE security functions and describes the physical and logical boundaries for the TOE.
- Security Environment (Section 3) – Describes the threats 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.
- Security Requirements (Section 5) – 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 6) – Describes the security functions provided by the TOE to satisfy the security requirements and objectives.
- Protection Profile Claims (Section 7) – Provides the identification of any ST Protection Profile claims as well as a justification to support such claims.
- Rationale (Section 8) – Presents the rationale for the security objectives, requirements, and the TOE summary specifications as to their consistency, completeness, and suitability.
- Acronyms (Section 9) – Defines the acronyms and terms used within this ST.

1.2 Security Target, TOE and CC Identification and Conformance

Table 1: ST, TOE, and CC Identification and Conformance

ST Title	Netezza Corporation Netezza Performance Server version 3.0, Release 6 [Build 6414] Netezza Security Target
ST Version	Version 1.1
Authors	Corsec Security, Inc. Jon Halperin and Matthew Appler
TOE Identification	Netezza Performance Server version 3.0, Release 6 [Build 6414]
Common Criteria (CC) Identification and Conformance	Common Criteria for Information Technology Security Evaluation, Version 2.3, [August 2005] (aligned with ISO/IEC 15408:2005); CC Part 2 conformant; CC Part 3 conformant; PP claim (none); Parts 2 and 3 Interpretations from the Interpreted CEM as of 9/4/2007 were reviewed, and no interpretations apply to the claims made in this ST.
PP Identification	None
Evaluation Assurance Level	EAL3+ (augmented with ALC_FLR.2, Flaw Reporting Procedures)
Keywords	Database, DBMS, Database Management, Data Warehousing

1.3 Conventions and Acronyms

1.3.1 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 several operations to be performed on security requirements: assignment, refinement, selection, and iteration. All of these operations are used within this ST. These operations are presented in the same manner in which they appear in Parts 2 and 3 of the CC with the following exceptions:

- 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.
- Iterations are identified by appending a letter in parenthesis 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.

1.3.2 Acronyms

The acronyms used within this ST are described in Section 9 – "Acronyms."

2 TOE Description

This section provides a general overview of the TOE as an aid to understanding the general capabilities and security requirements provided by the TOE. The TOE description provides a context for the TOE evaluation by identifying the product type and describing the evaluated configuration.

2.1 Product Type

The Netezza Performance Server (NPS) 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 NPS system 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 NPS product architecture.

The NPS is designed for databases ranging from approximately 2 terabytes to 100 terabytes, depending on the model chosen. The NPS 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.

2.2 Product Description

The NPS is a database appliance that integrates a database, server, and storage into a single system architecture. The architecture of the NPS database appliance is designed for query speed. Specifically, the NPS architecture is designed to allow efficient, ad-hoc querying of large amounts of data. This design of the NPS fundamentally alters the landscape for data warehousing and data analysis applications

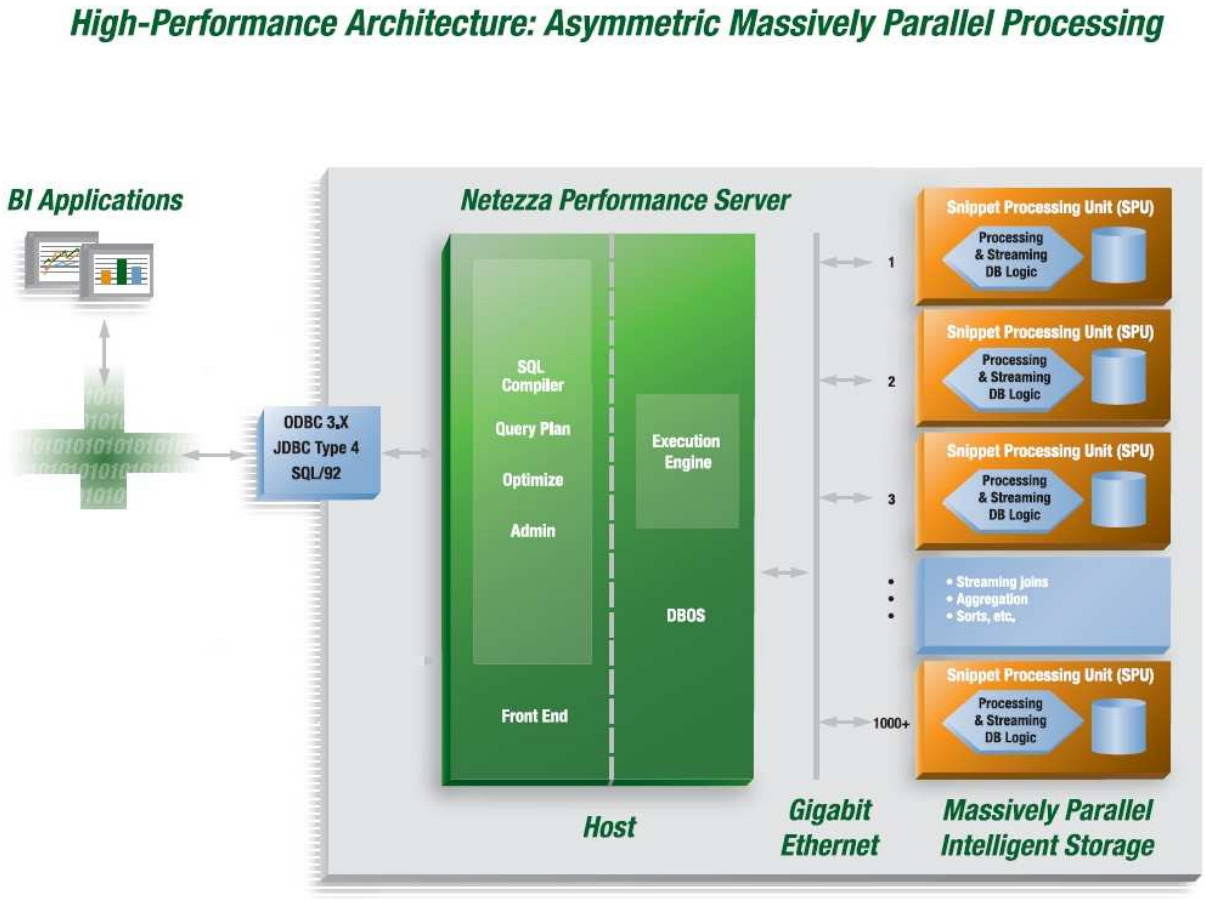
In a typical deployment of the NPS, data would be placed into the NPS from a corporate data source (e.g. an e-commerce transactional database, a corporate customer information database, or a corporate wide data collection system). Typically, end users of this product would then access this data through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

The Netezza Performance Server contains two primary components:

- **Host**
- **Snippet Processing Units (SPUs)**

These product components are deployed as shown in **Figure 1**.

Figure 1: NPS System



2.2.1 Host

The Host provides ODBC and JDBC connectivity, interfaces to system management applications, 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 or JDBC. After queries are received by the NPS, they are transformed from standard SQL to a query plan. 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 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 NPS system. As auditable events occur they are written to hard drives on the Host.

2.2.2 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 NPS, that data is distributed between 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 NPS 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 NPS system 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.

2.3 TOE Boundaries and Scope

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

2.3.1 Physical Boundary

Figure 2 illustrates the physical scope and the physical boundary of the overall solution and demonstrates the components of the TOE and the elements that constitute the TOE Environment.

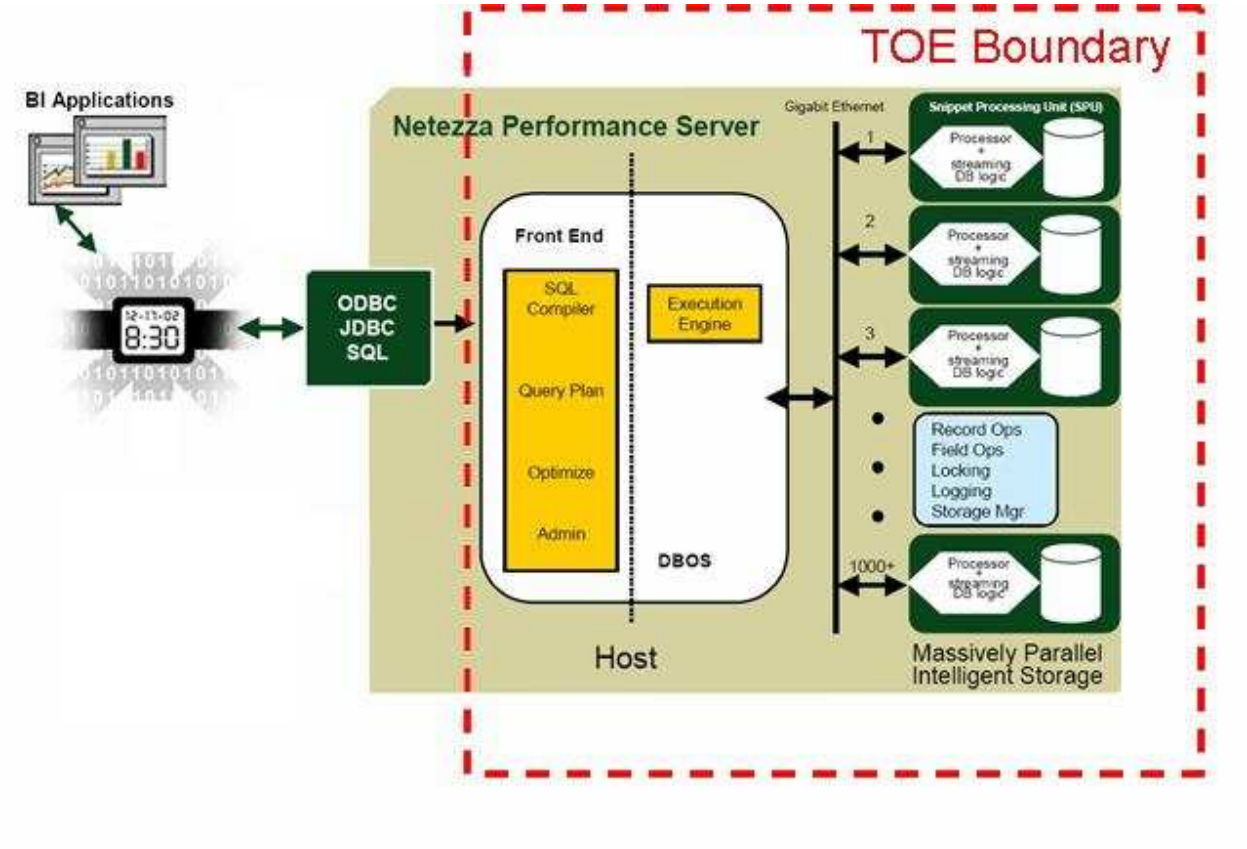


Figure 2: Physical TOE Boundary

The TOE consists of six hardware models running the NPS v3.0 software. The six models are the 5200, 8050z, 8150z, 8250z, 8450z, and 8650z.

The three primary physical components that comprise the TOE are:

- **Host:** The Host is the central intelligence component of the NPS architecture. It provides administrative functionality and interfaces with external entities.

- **Gigabit Ethernet Switch:** The Host and each SPU communicate via an internal network provided by this switch.
- **SPUs:** Each SPU consists of a hard drive and a processor. This is where low level processing of database queries occurs.

Other components within the TOE are:

- The operating system running on the Host: Red Hat Advanced Server 4.0
- Power supply
- KVM switch
- Host disk manager
- Host disk(s)

2.3.2 Logical Boundary

The security functional requirements implemented by the TOE are usefully grouped under the following Security Function Classes:

- Security Audit
- User Data Protection
- Identification and Authentication
- Security Management
- Protection of the TSF

2.3.2.1 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. Logs are kept which contain the records of regular operations and errors. The system audits numerous functions ranging from hardware failure 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, and the outcome of the event.

2.3.2.2 User Data Protection

User data protection defines how users of the NPS are allowed to perform operations on objects. The NPS is a database and all user data stored by the system is organized within individual database tables. The NPS provides a rich set of rights management 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 groups, thus providing a richer set of data rights management.

2.3.2.3 Identification and Authentication

All identification and authentication is managed by the Host component of the NPS. All users of the NPS are assigned a username and password. This username and password is then provided during the ODBC or JDBC 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 users 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.

2.3.2.4 Security Management

Security Management is provided on the NPS through the nzAdmin application. This application allows an administrator 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.

2.3.2.5 Protection of the TSF

The TOE protects itself by providing a domain for its own execution that cannot be accessed by untrusted subjects, and by ensuring that the TSFs cannot be bypassed. A TOE execution domain is provided by a combination of physical protection of the TOE, a TSF that prevents access by unauthorized users, and lack of visibility to non-TOE devices, users, or entities on the systems being monitored. Non-bypassability of the TSFs is provided by preventing unauthorized users access to the TOE and by enforcement of the access control mechanisms.

2.3.3 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.

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.

3 Security Environment

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.

3.1 Threats

This section identifies the threats to the IT assets against which protection is required by the TOE or by the security environment.

Table 2: Applicable Threats

Threat	Definition
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.
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.TSF_COMPROMISE	A malicious user or process may cause configuration data to be inappropriately accessed (viewed, modified, or deleted).
T.UNAUTHORIZED_ACCESS	A user may gain unauthorized access to user data for which they are not authorized according to the TOE security policy.
T.UNIDENTIFIED_ACTIONS	Failure of the authorized administrator to identify and act upon unauthorized actions may occur.

3.2 Organization Security Policies

An organizational security policy is a set of rules, practices, and procedures imposed by an organization to address its security needs.

Table 3: Applicable Policies

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

3.3 Assumptions

This section contains assumptions regarding the IT environment which the TOE will reside.

Table 4: Assumptions

Assumption	Definition
A.NO_EVIL	Administrators are non-hostile, appropriately trained, and follow all administrator guidance.
A.NO_GENERAL_PURPOSE	There are no general-purpose computing capabilities (e.g., compilers or user applications) available on 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.

4 Security Objectives

This section identifies the security objectives of the TOE and its supporting environment. The security objectives identify the responsibilities of the TOE and its environment in meeting the security needs.

4.1 TOE Security Objectives

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

Table 5: Security Objectives

Object name	Object Definition
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.AUDIT_REVIEW	The TOE will provide mechanisms to allow the authorized administrator to view and sort the audit logs.
O.AUDIT_STORAGE	The TOE will provide mechanisms to provide secure storage and management of the audit log.
O.CONFIGURATION_IDENTIFICATION	The configuration of the TOE is fully identified in a manner that will allow implementation errors to be identified, corrected with the TOE being redistributed promptly.
O.DOCUMENTED_DESIGN	The design of the TOE is adequately and accurately documented.
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.INTERNAL_TOE_DOMAINS	The TSF will maintain internal domains for separation of data and queries belonging to concurrent users.
O.NO_BYPASS	The TOE shall ensure that the TOE security mechanisms cannot be bypassed in order to gain access to the TOE resources.
O.PARTIAL_FUNCTIONAL_TEST	The TOE will undergo some security functional testing that demonstrates 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.TOE_ACCESS	The TOE will provide mechanisms that control a user's logical access to the TOE.

Object name	Object Definition
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.
O.I_AND_A	The TOE will contain identification and authentication mechanisms for users to login to the TOE.

4.2 Environment Security Objectives

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

Table 6: Environmental Security Objectives

Environmental Name Objective	Environmental Objective Definition
OE.NO_EVIL	Sites using the TOE shall ensure that authorized administrators are non-hostile, appropriately trained, and follow all administrator guidance.
OE.CONFIG	The TOE will be installed, configured, managed, and maintained in accordance with its guidance documentation and applicable security policies and procedures.
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.

5 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 1.3.1.

5.1 TOE Security Functional Requirements

This section contains the functional components from the Common Criteria (CC) Part 2 with the operations completed. For the conventions used in performing CC operations please refer to Section 1.3.1. A complete list of the SFRs met by the TOE is provided in Table 7.

Table 7: TOE Security Functional Requirements

Functional Components	
FAU_GEN.1	Audit data generation
FAU_GEN.2	User identity association
FAU_SAR.1	Audit Review
FAU_STG.1	Protected audit trail storage
FDP_ACC.1	Subset access control
FDP_ACF.1	Security attribute based access control
FIA_AFL.1	Authentication failure handling
FIA_ATD.1	User attribute definition
FIA_UAU.1	Timing of authentication
FIA_UID.1	Timing of identification
FIA_USB.1	User-subject binding
FMT_MOF.1	Management of security functions behavior
FMT_MSA.1	Management of security attributes
FMT_MSA.3	Static attribute initialization
FMT_MTD.1	Management of TSF data
FMT_REV.1	Revocation
FMT_SMF.1	Specification of management functions
FMT_SMR.1	Security roles
FPT_RVM.1	Non-bypassibility of the TSP
FPT_SEP.1	TSF domain separation
FPT_STM.1	Reliable time stamps

5.1.1 Class FAU: Security Audit

FAU_GEN.1 Audit Data Generation

Hierarchical to: No other components.

FAU_GEN.1.1

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

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

b) All auditable events for the [*minimum*] level of audit **listed in Table 8;** and

c) [*Start-up and shutdown of the DBMS*].

FAU_GEN.1.2

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

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

b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [*information specified in Table 8: Auditable* Events*, column three below*].

Table 8: Auditable Events

Security Functional Requirements	Auditable Event(s)	Additional Audit Record Contents
FAU_GEN.1	None	
FAU_GEN.2	None	
FDP_ACC.1	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.
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).	
FIA_ATD.1	None	
FIA_UAU.1	Unsuccessful use of the authentication mechanism	
FIA_UID.1	Unsuccessful use of the user identification mechanism, including the user identity provided.	
FIA_USB.1	Successful binding of user security attributes to a subject (e.g. adding a user to a group).	
FMT_MOF.1	All modifications in the behaviour of the functions in the TSF.	

Security Functional Requirements	Auditable Event(s)	Additional Audit Record Contents
FMT_MSA.1	All modifications of the values of security attributes.	
FMT_MSA.3	Modifications of the default setting of permissive or restrictive rules.	
FMT_MTD.1	All modifications to the values of the TSF data.	
FMT_REV.1	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_SEP.1	None	

Dependencies: FPT_STM.1 Reliable Time Stamps

FAU_GEN.2 User identity association

Hierarchical to: No other components.

FAU_GEN.2.1

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

Dependencies:

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

Application note: In some cases, an auditable event is not caused by a user, but is in response to an automated function. In this case, no user is associated with the auditable event.

FAU_SAR.1 Audit review

Hierarchical to: No other components.

FAU_SAR.1.1

The TSF shall provide [*authorized administrators*] with the capability to read [*all audit information*] from the audit records.

FAU_SAR.1.2

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

Dependencies: FAU_GEN.1 Audit data generation

FAU_STG.1 Protected audit trail storage

Hierarchical to: No other components.

FAU_STG.1.1

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

FAU_STG.1.2

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

Dependencies: FAU_GEN.1 Audit data generation

5.1.2 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 policy*] on [*all subjects, all DBMS-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 policy*] to objects based on the following:

- [*the authorized user identity associated with a subject, and*
- *access operations implemented for DBMS-controlled objects*].

FDP_ACF.1.2

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

[a) *If the requested mode of access is denied to that subject, deny access*.

b) *If the requested mode of access is permitted to that subject, permit access.*

c) *Else deny access*].

FDP_ACF.1.3

The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: [*No additional rules*].

FDP_ACF.1.4

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

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

5.1.3 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 Server*].

FIA_AFL.1.2

When the defined number of unsuccessful authentication attempts has been met or surpassed, 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:

[*Netezza Performance Server Group memberships; and*

Netezza Performance Server privileges].

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

FIA_UID.1 Timing of identification

Hierarchical to: No other components.

FIA_UID.1.1

The TSF shall allow [*user identification 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: [*Netezza Performance Server Group memberships, and Netezza Performance Server 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: [*none*].

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: [*only the authorized administrator can change security attributes*].

Dependencies: FIA_ATD.1 User Attribute Definition

5.1.4 Class FMT: Security Management

FMT_MOF.1 Management of security functions behavior

Hierarchical to: No other components.

FMT_MOF.1.1

The TSF shall restrict the ability to [*disable and enable*] the functions

[*review of audit records, and*

creation of database objects]

to [*authorized administrators*].

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

FMT_MSA.1 Management of security attributes

Hierarchical to: No other components.

FMT_MSA.1.1

The TSF shall enforce the [*Discretionary Access Control policy*] to restrict the ability to [*[manage]*] the security attributes [*of database users*] 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

FMT_MSA.3 Static attribute initialization

Hierarchical to: No other components.

FMT_MSA.3.1

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

FMT_MSA.3.2

The TSF shall allow the [*authorized administrator*] to specify alternative initial values to override the default values when an object or information is created.

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

FMT_MTD.1 Management of TSF data

Hierarchical to: No other components.

FMT_MTD.1.1

The TSF shall restrict the ability to [*modify*] the [*the groups and users that can interact with the TSF data*] to [*authorized administrators*].

Dependencies:

FMT_SMF.1 Specification of management functions

FMT_SMR.1 Security roles

FMT_REV.1 Revocation

Hierarchical to: No other components.

FMT_REV.1.1

The TSF shall restrict the ability to revoke security attributes associated with the [*users*] within the TSC to [*authorized administrators*].

FMT_REV.1.2

The TSF shall enforce the rules

[*Revocation rules will take effect at the beginning of the next attempt to access an object*].

Dependencies: FMT_SMR.1 Security roles

Application note: For example, if access permissions are changed on an object during a query that will not affect that query. However, if another query is attempted on the same object, the new permissions will then be enforced.

FMT_SMF.1 Specification of Management Functions

Hierarchical to: No other components.

FMT_SMF.1.1

The TSF shall be capable of performing the following security management functions: [*management of security functions, management of security attributes, management of TSF data*]

Dependencies: No Dependencies

FMT_SMR.1 Security roles

Hierarchical to: No other components.

FMT_SMR.1.1

Refinement: The TSF shall maintain the roles [*Admin account, Public group, and Administrator defined groups*].

FMT_SMR.1.2

The TSF shall be able to associate users with roles.

Dependencies: FIA_UID.1 Timing of identification

5.1.5 Class FPT: Protection of the TSF

FPT_RVM.1 Non-bypassability of the TSP

Hierarchical to: No other components.

FPT_RVM.1.1

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

Dependencies: No dependencies

FPT_SEP.1 TSF domain separation

Hierarchical to: No other components.

FPT_SEP.1.1

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

FPT_SEP.1.2

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

Dependencies: No Dependencies

FPT_STM.1 Reliable time stamps

FPT_STM.1.1

The TSF shall be able to provide reliable time stamps for its own use.

Dependencies: No Dependencies

5.2 Assurance Requirements

This ST contains all of the assurance requirements included in Evaluated Assurance Level (EAL) 3 augmented with the following additions:

- ALC_FLR.2: Flaw reporting procedures

Assurance requirements listed in Table 9, are needed for Basic Robustness. These requirements are taken from the CC Part 3 and are summarized below.

Table 9: Assurance Requirements

Assurance Class	Assurance Component	
Configuration Management	ACM_CAP.3	Authorization Controls
	ACM.SCP.1	TOE CM Coverage
Delivery and Operation	ADO_DEL.1	Delivery procedures
	ADO_IGS.1	Installation, generation, and start-up procedures
Development	ADV_FSP.1	Informal functional specification
	ADV_HLD.2	Security enforcing high-level design
	ADV_RCR.1	Informal correspondence demonstration
Guidance Documents	AGD_ADM.1	Administrator guidance
	AGD_USR.1	User guidance
Life Cycle Support	ALC_FLR.2	Flaw reporting procedures
	ALC_DVS.1	Identification of security measures
Tests	ATE_COV.2	Analysis of coverage
	ATE_DPT.1	Testing: high-level design
	ATE_FUN.1	Functional testing
	ATE_IND.2	Independent testing – sample
Vulnerability Assessment	AVA_MSU.1	Examination of guidance
	AVA_SOF.1	Strength of TOE security functional evaluation
	AVA_VLA.1	Developer vulnerability analysis

6 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.

6.1 TOE Security Functions

Each of the security function descriptions is organized by the security requirements corresponding to the security function. 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 are suitable to satisfy the necessary requirements.

Table 10: 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_STG.1	Protected audit trail storage
User Data Protection	FDP_ACC.1	Subset access control
	FDP_ACF.1	Security attribute based access control
Identification and Authentication	FIA_AFL.1	Authentication Failure Handling
	FIA_ATD.1	User attribute definition
	FIA_UAU.1	Timing of authentication
	FIA_UID.1	Timing of identification
	FIA_USB.1	User-subject binding
Security Management	FMT_MOF.1	Management of security functions behavior
	FMT_MSA.1	Management of security attributes
	FMT_MSA.3	Static attribute initialization
	FMT_MTD.1	Management of TSF data
	FMT_REV.1	Revocation
	FMT_SMF.1	Specification of management functions
	FMT_SMR.1	Security roles
Protection of the TSF	FPT_RVM.1	Non-bypassability of the TSP
	FPT_SEP.1	TSF domain separation
	FPT_STM.1	Reliable time stamps

6.1.1 Security Audit

The TOE generates two types of audit data. The first type contains information about authentication, access control and event handling and the second type records user activity with the database. Audit logs are generated solely on the Host and stored in the file system of the Host operating system.

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 NPS 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 was 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 NPS system:

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.

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

6.1.2 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[1] within a given database.

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".

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.

All user data stored by the TOE exists as a database Object. This can take several forms, e.g., 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.

TOE Security Functional Requirements Satisfied: FDP_ACC.1, FDP_ACF.1

6.1.3 Identification and Authentication

There are two identification and authentication mechanisms used by the TOE. A user may be required to authenticate to the Linux Operating System (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.

6.1.3.1 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.

6.1.3.2 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 or JDBC API. Over each of these interfaces the user is required to provide a username and password prior to gaining access to system services.

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.

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

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 TOE makes a claim of SOF-basic; this claim is in reference to the security mechanisms provided by FIA_UAU.1.

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

6.1.4 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 Administrator defined groups role is defined as every other user account available on the TOE.

All access rights within the TOE are granted based upon the User Data Protection mechanisms provided through the Access Control Matrix (ACM). The privileges that can be assigned through this ACM mechanism are described in more detail in section 6.1.2.

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, FMT_MSA.1, FMT_MSA.3, FMT_MTD.1, FMT_REV.1, FMT_SMF.1, FMT_SMR.1].

6.1.5 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 SPU's. Protection of the TOE from physical tampering is ensured by its environment. It is the responsibility of the administrator to assure that the physical connections made to the TOE remain intact and unmodified. The TOE is self contained; the hardware and firmware provided by the NPS system provide all the services necessary to implement the TOE. There are no other external interfaces into the TOE other than the Ethernet interfaces. No general purpose operating system, programming interfaces or external disk storage is provided.

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 OS. From there, other subsystems are able to retrieve the time for inclusion in audit records.

The TOE maintains a security domain for its own execution that protects it from interference and tampering by untrusted subjects. Software files on the TOE cannot be modified without violating the physical security of the TOE. The underlying assumption regarding the operation of the TOE is that it is maintained in a physically secure environment.

Non-bypassability of the TOE is provided by a combination of the basic configuration and enforcement of the security policy rules. The assumed secure basic configuration maintaining physical and logical isolation supports the protection of Security Functions. The functions that enforce the TOE Security Policy (TSP) will always be invoked, before any function within the TSF Scope of Control is allowed to proceed.

TOE Security Functional Requirements Satisfied: [FPT_RVM.1, FPT_SEP.1, FPT_STM.1,].

6.2 TOE Security Assurance Measures

EAL3+ was chosen to provide a basic level of independently assured security. This section of the Security Target maps the assurance requirements of the TOE for a CC EAL3+ level of assurance to the assurance measures used for the development and maintenance of the TOE. The following table provides a mapping of the appropriate documentation to the TOE assurance requirements.

Note to Evaluator: The final versions of these documents have not yet been produced. The version numbers will be completed when the evaluation is close to completion and the documents have been finalized.

Table 11: Assurance Measures Mapping to TOE Security Assurance Requirements (SARs)

Assurance Component	Assurance Measure
ACM_CAP.3	Netezza Performance Server version 3.0, Release 6 [Build 6414] - Configuration Management v0.1
ACM.SCP.1	Netezza Performance Server version 3.0, Release 6 [Build 6414] - Configuration Management v0.1
ADO_DEL.1	Netezza Performance Server version 3.0, Release 6 [Build 6414] - Secure Delivery v0.1
ADO_IGS.1	Netezza Performance Server 2.5 Standard System Configuration Guide v1.0
	Netezza Performance Server 3.0 Getting Started Tips v1.0
ADV_FSP.1	Netezza Performance Server version 3.0, Release 6 [Build 6414] - TOE Architecture: Functional Specification v0.1
ADV_HLD.2	Netezza Performance Server version 3.0, Release 6 [Build 6414] - TOE Architecture: High Level Design v0.1
ADV_RCR.1	Netezza Performance Server version 3.0, Release 6 [Build 6414] Representation Correspondence v0.1
AGD_ADM.1	Netezza Performance Server version 3.0, Release 6 [Build 6414] Administrator's Guide v 1.0
AGD_USR.1	Not Applicable
ALC_DVS.1	Netezza Performance Server version 3.0, Release 6 [Build 6414] –ALC-DVS.1 v0.1
ALC_FLR.2	Netezza Performance Server version 3.0, Release 6 [Build 6414]- ALC_FLR.2 v0.1.doc
ATE_COV.2	Netezza Performance Server version 3.0, Release 6 [Build 6414] – Functional Tests and Coverage v0.1
ATE_DPT.1	Netezza Performance Server version 3.0, Release 6 [Build 6414] – Functional Tests and Coverage v0.1
ATE_FUN.1	Netezza Performance Server version 3.0, Release 6 [Build 6414] – Functional Tests and Coverage v0.1
ATE_IND.2	Provided by the CC Evaluation Lab.
AVA_MSU.1	Netezza Performance Server version 3.0, Release 6 [Build 6414] - Vulnerability Assessment v0.1

Assurance Component	Assurance Measure
AVA_SOF.1	Netezza Performance Server version 3.0, Release 6 [Build 6414] - Vulnerability Assessment v0.1
AVA_VLA.1	Netezza Performance Server version 3.0, Release 6 [Build 6414] - Vulnerability Assessment v0.1

6.2.1 ACM_CAP.3: Configuration Management Document ACM_SCP.1: Scope

The Configuration Management document provides a description of the various tools used to control the configuration items and how they are used internally at Netezza. This document provides a complete configuration item list and a unique referencing scheme for each configuration item. Additionally, the configuration management system is described including procedures that are used by developers to control and track changes that are made to the TOE. The documentation further details the TOE configuration items that are controlled by the configuration management system.

6.2.2 ADO_DEL.1: Delivery and Operation Document

The Delivery and Operation document provides a description of the secure delivery procedures implemented by Netezza to protect against TOE modification during product delivery. The Installation Documentation provided by Netezza details the procedures for installing the TOE and placing the TOE in a secure state offering the same protection properties as the master copy of the TOE. The Installation Documentation provides guidance to the TOE User(s) on configuring the TOE and how they affect the TSF.

6.2.3 ADO_IGS.1: Installation Guidance, AGD_ADM.1: Administrator Guidance, AGD_USR.1: User Guidance

The installation guidance document provides the procedures necessary for the secure installation, generation, and start-up of the TOE for administrators and users of the TOE.

The administrator guidance documentation provides detailed procedures for the administration of the TOE and description of the security functions provided by the TOE.

The User Guidance documentation provided directs users on how to operate the TOE in a secure manner. Additionally, User Guidance explains the user-visible security functions and how they need to be exercised.

6.2.4 ADV_FSP.1: Informal Functional Specification, ADV_HLD.2: High Level Design, ADV_RCR.1: Representation Correspondence

The Netezza design documentation consists of several related design documents that address the components of the TOE at different levels of abstraction. The following design documents address the Development Assurance Requirements:

- The Functional Specification provides a description of the security functions provided by the TOE and a description of the external interfaces to the TSF. The Functional Specification covers the purpose and method of use and a list of effects, exceptions, and errors message for each external TSF interface.
- The High-Level Design provides a top level design specification that refines the TSF functional specification into the major constituent parts (subsystems) of the TSF. The high-level design identifies the basic structure of the TSF, the major elements, a listing of all interfaces, and the purpose and method of use for each interface.
- The Representation Correspondence demonstrates the correspondence between each of the TSF representations provided. This mapping is performed to show the functions traced from the ST description to the High-Level Design.

6.2.5 ALC_DVS.1: Development Security, ALC_FLR.2: Flaw Remediation

The Life Cycle Support documentation describes all the physical, procedural, personnel, and other security measures that are necessary to protect the confidentiality and integrity of the TOE design and implementation in its development environment. It provides evidence that these security measures are followed during the development and maintenance of the TOE. Flaw remediation procedures addressed to TOE developers are provided and so are the established procedures for accepting and acting upon all reports of security flaws and requests for corrections of those flaws. Flaw remediation guidance addressed to TOE users is provided. The description also contains the procedures used to track all reported security flaws in each release of the TOE. The established life-cycle model to be used in the development and maintenance of the TOE is documented and explanation on why the model is used is also documented. The selected implementation-dependent options of the development tools are described.

6.2.6 ATE_COV.2: Test Coverage Analysis, ATE_FUN.1: Functional Testing, ATE_DPT.1: Depth of Coverage Analysis

There are a number of components that make up the Test documentation. The Coverage Analysis demonstrates that testing is performed against the functional specification. The Coverage Analysis demonstrates the extent to which the TOE security functions were tested as well as the level of detail to which the TOE was tested. Test Plans and Test Procedures, which detail the overall efforts of the testing effort and break down the specific steps taken by a tester, are also provided in order to meet the assurance requirement Functional Testing.

6.2.7 AVA_VLA.1: Vulnerability Analysis, AVA_SOF.1: Strength of Function Analysis

A Vulnerability Assessment is provided to demonstrate ways in which an entity could violate the TSP and provide a list of identified vulnerabilities. Additionally, this document provides evidence of how the TOE is resistant to obvious attacks.

The Strength of TOE Security Function Analysis demonstrates the strength of the probabilistic or permutational mechanisms employed to provide security functions within the TOE and how they exceed the minimum SOF requirements.

7 Protection Profile Claims

This section provides the identification and justification for any Protection Profile conformance claims.

7.1 Protection Profile Reference

There are no protection profile claims for this security target.

8 Rationale

This section provides the rationale for the selection of the security requirements, objectives, assumptions, and threats. In particular, it shows that the security requirements are suitable to meet the security objectives, which in turn are shown to be suitable to cover all aspects of the TOE security environment.

8.1 Security Objectives Rationale

This section provides a rationale for the existence of each assumption, threat, and policy statement that compose the Security Target. Table 12 demonstrates the mapping between the assumptions, threats, and polices to the security objectives is complete. The following discussion provides detailed evidence of coverage for each assumption, threat, and policy.

Table 12: Relationship of Security Threats to Objectives

Threats and Policies / Objectives	O.ADMIN_GUIDAN	O.ADMIN_ROLE	O.AUDIT_GENERA	O.AUDIT_REVIEW	O.AUDIT_STORAG	O.CONFIGURATIO	O.DOCUMENTED	O.MANAGE	O.MEDIATE	O.INTERNAL_TOE	O.PARTIAL_SELF	O.TOE_ACCESS	O.VULNERABILITY	O.PARTIAL_FUNC	O.I_AND_A	O.NO_BYPASS
T. ACCIDENTAL_ADMIN_ERROR	✓															
T.MASQUERADE												✓			✓	
T.POOR_DESIGN					✓	✓	✓						✓			✓
T.POOR_IMPLEMENTATION						✓					✓		✓	✓		✓
T.POOR_TEST							✓				✓		✓	✓		
T.TSF_COMPROMISE								✓		✓	✓					
T.UNAUTHORIZED_ACCESS									✓						✓	
T.UNIDENTIFIED_ACTIONS	✓		✓					✓								
P.ACCOUNTABILITY			✓	✓								✓				
P.ROLES		✓														

T. ACCIDENTAL_ADMIN_ERROR

An administrator may incorrectly install or configure the TOE resulting in ineffective security mechanisms.

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 in a way that is insecure.

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 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. O.I_AND_A helps to mitigate this threat by providing for mechanisms to identify and authenticate users.

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 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 ensures that the design of the TOE is documented, permitting detailed review by evaluators and validators. O.VULNERABILITY_ANALYSIS ensures the design of the TOE is analyzed for design flaws. O.AUDIT_STORAGE ensures that the audit logs are securely stored and managed, and O.NO_BYPASS ensures that poor design does not result in a design flaw that allows the TSP to be bypassed.

T.POOR_IMPLEMENTATION

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

O.CONFIGURATION_IDENTIFICATION plays a role in countering this treat 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 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 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. 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. O.PARTIAL_SELF_PROTECTION helps reduce the availability of vulnerabilities from untrusted users. O.NO_BYPASS ensures that poor implementation does not result in a configuration that allows the TSP to be bypassed.

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 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 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 addresses this concern by requiring a vulnerability analysis be performed in conjunction with testing that goes beyond functional testing. This objective provides a measure of confidence that the TOE does not contain security flaws that may not be identified through functional testing. While these testing activities are a necessary activity for successful

completion of an evaluation, this testing activity does not address the concern that the TOE continues to operate correctly and enforce its security policies once it has been fielded. Some level of testing must be available to end users to ensure the TOE's security mechanisms continue to operator correctly once the TOE is fielded. O.PARTIAL_SELF_PROTECTION helps reduce the availability of vulnerabilities from untrusted users.

T.TSF_COMPROMISE

A malicious user or process may cause configuration data to be inappropriately accessed (viewed, modified, or deleted).

O.PARTIAL_SELF_PROTECTION ensures the TOE is capable of protecting itself from attack. 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.INTERNAL_TOE_DOMAINS ensures the TOE will establish separate domains for data belonging to users.

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.MEDIATE ensures that all accesses to user data are subject to mediation, unless said data has been specifically identifies as public data. The TOE requires successful authentication to the TOE prior to gaining access to any controlled-access content. By implementing strong authentication to gain access to these services, an attacker's opportunity to successfully conduct a man-in-the-middle and/or password guessing attack 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. O.I_AND_A ensures that only authorized users may access the TOE.

T.UNIDENTIFIED_ACTIONS

Failure of the authorized administrator to identify and act upon unauthorized actions may occur.

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). 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 review audit records. O.AUDIT_GENERATION ensures that the authorized administrators have a means of identifying unusual activity.

P.ACCOUNTABILITY

The authorized users of the TOE shall be held accountable for their actions within the TOE.

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. O.AUDIT_REVIEW provides authorized administrators with the ability to review the audit trail. 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.). The audit mechanism is required to include the current date and time in each audit

record. All audit records that include the user ID, will also include the date and time that the event occurred. O.TOE_ACCESS supports this policy by requiring the TOE to identify and authenticate all authorized users prior to allowing any TOE access or any TOE mediated access on behalf of those users.

P.ROLES

The TOE shall provide an authorized administrator role for secure administration of the TOE. This role shall be separate and distinct from other authorized users.

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).

Table 13: Relationship of Environmental Objectives to Assumptions

Assumptions \ Objectives	OE.NO_EVIL	OE.CONFIG	OE. NO_GENERAL_PURPOSE	OE.PHYSICAL
A.NO_EVIL	✓	✓		
A.NO_GENERAL_PURPOSE			✓	
A.PHYSICAL				✓

A.NO_EVIL

Administrators are non-hostile, appropriately trained, and follow all administrator guidance.

OE.NO_EVIL specifies that sites using the TOE shall ensure that authorized administrators are non-hostile, appropriately trained, and follow all administrator guidance. OE.CONFIG ensures that the TOE will be installed, configured, managed, and maintained in accordance with its guidance documentation and applicable security policies and procedures.

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 states that 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 states that 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.

8.2 Security Functional Requirements Rationale

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

Table 14: Relationship of Security Requirements to Objectives

	Objectives / Requirements	O.ADMIN_GUIDANCE	O.ADMIN_ROLE	O.AUDIT_GENERATION	O.AUDIT_REVIEW	O.AUDIT_STORAGE	O.CONFIGURATION_IDENTIFICATION	O.DOCUMENTED_DESIGN	O.MANAGE	O.MEDIATE	O.INTERNAL_TOE_DOMAINS	O.PARTIAL_SELF_PROTECTION	O.TOE_ACCESS	O.VULNERABILITY_ANALYSIS	O.PARTIAL_FUNCTIONAL_TEST	O.NO_BYPASS	O.I_AND_A
TOE	FAU_GEN.1			✓													
	FAU_GEN.2			✓													
	FAU_SAR.1				✓												
	FAU_STG.1					✓											
	FDP_ACC.1									✓							
	FDP_ACF.1									✓							
	FIA_AFL.1																✓
	FIA_ATD.1												✓				
	FIA_UAU.1																✓
	FIA_UID.1																✓
	FIA_USB.1																✓
	FMT_MOF.1								✓								
	FMT_MSA.1								✓								
	FMT_MSA.3								✓								
	FMT_MTD.1								✓								
	FMT_REV.1								✓								
	FMT_SMF.1								✓								
	FMT_SMR.1		✓						✓								
	FPT_RVM.1														✓		
	FPT_SEP.1										✓	✓					
	FPT_STM.1			✓													

Objectives / Requirements	O.ADMIN_GUIDANCE	O.ADMIN_ROLE	O.AUDIT_GENERATION	O.AUDIT_REVIEW	O.AUDIT_STORAGE	O.CONFIGURATION_IDENTIFICATION	O.DOCUMENTED_DESIGN	O.MANAGE	O.MEDIATE	O.INTERNAL_TOE_DOMAINS	O.PARTIAL_SELF_PROTECTION	O.TOE_ACCESS	O.VULNERABILITY_ANALYSIS	O.PARTIAL_FUNCTIONAL_TEST	O.NO_BYPASS	O.I_AND_A
ATE_COV.2														✓		
ATE_FUN.1														✓		
ATE_IND.2														✓		
ADO_IGS.1	✓															
AGD_ADM.1	✓															
AGD_USR.1	✓															
AVA_MSU.1	✓															
ADO_DEL.1	✓															
ALC_FLR.2						✓										
ADV_FSP.1							✓									
ADV_HLD.1							✓									
ADV_RCR.1							✓									
AVA_SOF.1												✓				
AVA_VLA.1													✓			
ACM_CAP.3						✓										

O.ADMIN_GUIDANCE

The TOE will provide administrators with the necessary information for secure management.

ADO_DEL.1 ensures that the administrator is provided documentation that instructs them how to ensure the delivery of the TOE, in whole or in parts, has not been tampered with or corrupted during delivery. This requirement ensures the administrator has the ability to begin their TOE installation with a clean (e.g., malicious code has not been inserted once it has left the developer's control) version of the TOE, which is necessary for secure management of the TOE. ADO_IGS.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 Installation, Generation, and Startup (IGS) documentation ensures that once the administrator has followed the installation and configuration guidance the result is a

TOE in a secure configuration. AGD_ADM.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 description of how to setup and review the auditing features of the TOE. AGD_USR.1 is intended for non-administrative users, but could be used to provide guidance on security that is common to both administrators and non-administrators (e.g., password management guidelines). AVA_MSU.1 ensures that the guidance documentation is complete and consistent, and notes all requirements for external security measures.

O.ADMIN_ROLE

The TOE will provide authorized administrator roles to isolate administrative actions.

The TOE will establish, at least, an authorized administrator role. The ST writer may choose to specify more roles. The authorized administrator will be given privileges to perform certain tasks that other users will not be able to perform. These privileges include, but are not limited to, access to audit information and security functions. (FMT_SMR.1)

O.AUDIT_GENERATION

The TOE will provide the capability to detect and create records of security relevant events associated with users.

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 takes place in the TOE. This requirement also defines the information that must be contained in the audit record for each auditable event. FAU_GEN.2 ensures that the audit records associate a user identity with the auditable event. In the case of authorized users, the association is accomplished with the userid. FPT_STM.1 ensures that reliable timestamps are available to be included in the audit records.

O.AUDIT_REVIEW

The TOE will provide mechanisms to allow the authorized administrator to view and sort the audit logs.

FAU_SAR.1 addresses the capability to allow authorized administrators to review the audit logs. The records must be presented in a manner suitable for interpretation.

O.AUDIT_STORAGE

The TOE will provide mechanisms to provide secure storage and management of the audit log.

FAU_STG.1 requires that only an authorized administrator may delete the audit records, ensuring that malicious users may not compromise the data stored within the audit records.

O.CONFIGURATION_IDENTIFICATION

The configuration of the TOE is fully identified in a manner that will allow implementation errors to be identified, corrected with the TOE being redistributed promptly.

ACM_CAP.3 addresses this objective by requiring that there be a unique reference for the TOE, and that the TOE is labeled with that reference. It also requires that there be a CM system in place, and that the configuration items that comprise the TOE are uniquely identified. This provides a clear identification of the composition of the TOE. ALC_FLR.2 addresses this objective by

requiring that there be a mechanism in place for identifying flaws subsequent to fielding, and for distributing those flaws to entities operating the system.

O.DOCUMENTED_DESIGN

The design of the TOE is adequately and accurately documented.

ADV_FSP.1 requires that the interfaces to the TOE be documented and specified.ADV_HLD.1 requires the high level design of the TOE be documented and specified and that said design be shown to correspond to the interfaces. ADV_RCR.1 requires that there be a correspondence between adjacent layers of the design decomposition.

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 requires that the ability to use particular TOE capabilities be restricted to the administrator. FMT_MSA.1 requires that the ability to perform operations on security attributes be restricted to particular roles. FMT_MSA.3 requires that default values used for security attributes are restrictive, and that the administrator has the ability to override those values. FMT_MTD.1 requires that the ability to manipulate TOE content is restricted to administrators. FMT_REV.1 restricts the ability to revoke attributes to the administrator. FMT_SMF.1 identifies the management functions that are available to the authorized administrator. 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 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 operation between subject and object covered are defined by the TOE's policy. FDP_ACF.1 defines the security attribute used to provide access control to objects based on the TOE's access control policy.

O.INTERNAL_TOE_DOMAINS

The TSF will maintain internal domains for separation of data and queries belonging to concurrent users.

FPT_SEP.1 requires the TOE to maintain a separate domain for its own execution separate from other processes.

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.TOE_ACCESS

The TOE will provide mechanisms that control a user's logical access to the TOE.

FIA_ATD.1 defines the attributes of users. This includes the privileges and group memberships associated with that user. These, combined with the object privileges are used to autogenerate a list of effective privileges which provide a mechanism for performing access checks and retrieving lists of accessible objects. This is used by the TOE to enforce user access to the TOE data. AVA_SOF.1 requirement is applied to the password mechanism used by the local administrator

(The single use authentication mechanism supplies by the IT environment (i.e., authentication server) has this same assurance requirement levied against it to ensure a consistent level of assurance.) For this TOE, the strength of function specified is basic. This requirement ensures the developer has performed an analysis of the password mechanism to ensure the probability of guessing a local administrator's password would require a high-attack potential, as defined in Annex B of the CEM. This analysis takes into account the password spaces, as well as any feature of the password mechanism that plays a role in limiting the number of failed authentication attempts within a given time period.

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.

The AVA_VLA.1 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_VLA.1 requires the developer to perform a systematic search for potential vulnerabilities in all the TOE deliverables. For those vulnerabilities that are not eliminated, a rationale must be provided that describes why these vulnerabilities cannot be exploited by a threat agent with a low attack potential, which is in keeping with the desired assurance level of this TOE. As with the functional testing, a key element is this component is that an independent assessment of the completeness of the developer's analysis is made, and more importantly, an independent vulnerability analysis coupled with testing of the TOE is performed. This component provides the confidence that security flaws do not exist in the TOE that could be exploited by a threat agent or moderate (or lower) attack potential to violate the TOE's security policies.

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 requires that there be a correspondence between the tests in the test documentation and the TSF as described in the functional specification. ATE_FUN.1 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 test performed, and test scenarios. These require that the developer run those tests, and show that the expected results were achieved. 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.NO_BYPASS

The TOE shall ensure that security mechanisms cannot be bypassed in order to gain access to the TOE resources.

FPT_RVM.1 ensures the TOE cannot be bypassed in order to gain unauthorized access of TOE resources.

O.I_AND_A

The TOE contains identification and authentication mechanisms for users to login to the TOE.

FIA_AFL.1 ensures 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_UAU.1 requires that all users must authenticate before they are given access to the TOE. FIA_UID.1 requires that users must uniquely identify themselves before they are given access to the TOE. FIA_USB.1 binds the user identity with group memberships and privileges to determine access to data.

8.2.1 Rationale for the IT Environment

OE.NO_EVIL

Sites using the TOE shall ensure that authorized administrators are non-hostile, appropriately trained, and follow all administrator guidance.

This objective does not contain any IT security requirements because it is a non-IT related objective. Thus, the CC does not mandate it map to any requirements.

OE.CONFIG

The TOE will be installed, configured, managed, and maintained in accordance with its guidance documentation and applicable security policies and procedures.

This objective does not contain any IT security requirements because it is a non-IT related objective. Thus, the CC does not mandate it map to any requirements.

OE. NO_GENERAL_ PURPOSE

There will be no general-purpose computing capabilities (e.g., compilers or user applications) available on DMBS servers, other than those services necessary for the operation, administration, and support of the DBMS.

This objective does not contain any IT security requirements because it is a non-IT related objective. Thus, the CC does not mandate it map to any requirements.

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.

This objective does not contain any IT security requirements because it is a non-IT related objective. Thus, the CC does not mandate it map to any requirements.

8.3 Dependency Rationale

This ST does satisfy all the requirement dependencies of the Common Criteria. Table 15 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 15: Functional Requirements Dependencies

Requirement	Dependency	Dependency Met	Rationale
FAU_GEN.1	FPT_STM.1	[✓]	Satisfied by the IT environment with FPT_STM.1.
FAU_GEN.2	FAU_GEN.1 FIA_UID.1	[✓]	Satisfied.
FAU_SAR.1	FAU_GEN.1	[✓]	Satisfied.
FAU_STG.1	FAU_GEN.1	[✓]	Satisfied.
FIA_AFL.1	FIA_UAU.1	[✓]	Satisfied.
FIA_UAU.1	FIA_UID.1	[✓]	Satisfied.
FIA_USB.1	FIA_ATD.1	[✓]	Satisfied.

Requirement	Dependency	Dependency Met	Rationale
FDP_ACC.1	FDP_ACF.1	[✓]	Satisfied.
FDP_ACF.1	FDP_ACC.1 FMT_MSA.3	[✓]	Satisfied.
FMT_MOF.1	FMT_SMF.1 FMT_SMR.1	[✓]	Satisfied.
FMT_MSA.1	[FDP_ACC.1 or FDP_IFC.1] FMT_SMF.1 FMT_SMR.1	[✓]	Dependency satisfied by FDP_ACC.1.
FMT_MSA.3	FMT_MSA.1 FMT_SMR.1	[✓]	Satisfied.
FMT_MTD.1	FMT_SMF.1 FMT_SMR.1	[✓]	Satisfied.
FMT_REV.1	FMT_SMR.1	[✓]	Satisfied.
FMT_SMR.1	FIA_UID.1	[✓]	Satisfied.

8.4 TOE Summary Specification Rationale

8.4.1 TOE Summary Specification Rationale for the Security Functional Requirements

Each subsection in the TOE Summary Specification (Section 6) describes a security function of the TOE. Each description is organized by set of requirements with rationale that indicates how these requirements are satisfied by aspects of the corresponding security function. The set of security functions work together to satisfy all of the security functions and assurance requirements. Furthermore, all of the security functions are necessary in order for the TSF to provide the required security functionality. This section, in conjunction with the TOE Summary Specification section, provides evidence that the security functions are suitable to fulfill the TOE security requirements.

Table 16 identifies the relationship between security requirements and security functions, showing that all security requirements are addressed and all security functions are necessary (i.e., they correspond to at least one security requirement).

The only security mechanism that is realized by a probabilistic or permutational implementation is the password mechanism. For an analysis of the Strength of Function, refer to Strength of Function (SOF) Rationale section.

Table 16: Mapping of Security Functional Requirements to TOE Security Functions

TOE Security Function	SFR	Rationale
Security Audit	FAU_GEN.1	Audit data generation
	FAU_GEN.2	User identity association
	FAU_SAR.1	Audit review
	FAU_STG.1	Protected audit trail storage
User Data Protection	FDP_ACC.1	Subset access control
	FDP_ACF.1	Security attribute based access control
Identification and Authentication	FIA_AFL.1	Authentication failure handling
	FIA_ATD.1	User attribute definition

TOE Security Function	SFR	Rationale
	FIA_UAU.1	Timing of authentication
	FIA_UID.1	Timing of identification
	FIA_USB.1	User-subject binding
Security Management	FMT_MOF.1	Management of security functions behavior
	FMT_MSA.1	Management of security attributes
	FMT_MSA.3	Static attribute initialization
	FMT_MTD.1	Management of TSF data
	FMT_REV.1	Revocation
	FMT_SMF.1	Specification of management functions
	FMT_SMR.1	Security roles
Protection of the TSF	FPT_RVM.1	Non-bypassability of the TSP
	FPT_SEP.1	TSF domain separation
	FPT_STM.1	Reliable time stamps

8.4.2 TOE Summary Specification Rationale for the Security Assurance Requirements

EAL3+ was chosen to provide a basic level of independently assured security in the absence of ready availability of the complete development record from the vendor. The chosen assurance level is consistent with the postulated threat environment.

8.4.2.1 Configuration Management

The Configuration Management documentation provides a description of tools used to control the configuration items and how they are used at the Netezza. The documentation provides a complete configuration item list and a unique reference for each item. Additionally, the configuration management system is described including procedures that are used by developers to control and track changes that are made to the TOE. The documentation further details the TOE configuration items that are controlled by the configuration management system.

Corresponding CC Assurance Components:

- Configuration Items

8.4.2.2 Delivery and Operation

The Delivery and Operation documentation provides a description of the secure delivery procedures implemented by Netezza to protect against TOE modification during product delivery. The Installation Documentation provided by Netezza details the procedures for installing the TOE and placing the TOE in a secure state offering the same protection properties as the master copy of the TOE. The Installation Documentation provides guidance to the administrator on the TOE configuration parameters and how they affect the TSF.

Corresponding CC Assurance Components:

- Delivery Procedures
- Installation, Generation and Start-Up Procedures

8.4.2.3 Development

The Netezza design documentation consists of several related design documents that address the components of the TOE at different levels of abstraction. The following design documents address the Development Assurance Requirements:

- The Functional Specification provides a description of the security functions provided by the TOE and a description of the external interfaces to the TSF. The Functional Specification covers the purpose and method of use and a list of effects, exceptions, and errors message for each external TSF interface.
- The High-Level Design provides a top level design specification that refines the TSF functional specification into the major constituent parts (subsystems) of the TSF. The high-level design identifies the basic structure of the TSF, the major elements, a listing of all interfaces, and the purpose and method of use for each interface.
- The Correspondence Analysis demonstrates the correspondence between each of the TSF representations provided. This mapping is performed to show the functions traced from the ST description to the High-Level Design.

Corresponding CC Assurance Components:

- Informal Functional Specification
- Descriptive High-Level Design
- Informal Representation Correspondence

8.4.2.4 Guidance Documentation

The Netezza Guidance documentation provides administrator and user guidance on how to securely operate the TOE. The administrator Guidance provides descriptions of the security functions provided by the TOE. Additionally, it provides detailed accurate information on how to administer the TOE in a secure manner and how to effectively use the TSF privileges and protective functions. The User Guidance provided directs users on how to operate the TOE in a secure manner. Additionally, User Guidance explains the user-visible security functions and how they are to be used and explains the user's role in maintaining the TOE's Security. Netezza provides single versions of documents which address the administrator Guidance and User Guidance; there are not separate guidance documents specifically for non-administrator users of the TOE.

Corresponding CC Assurance Components:

- Administrator Guidance
- User Guidance

8.4.2.5 Tests

There are a number of components that make up the Test documentation. The Coverage Analysis demonstrates the testing performed against the functional specification. The Coverage Analysis demonstrates the extent to which the TOE security functions were tested as well as the level of detail to which the TOE was tested. Netezza Test Plans and Test Procedures, which detail the overall efforts of the testing effort and break down the specific steps taken by a tester, are also provided.

Corresponding CC Assurance Components:

- Analysis of Coverage
- Testing: high level design
- Functional Testing

8.4.2.6 Vulnerability and TOE Strength of Function Analyses

A Vulnerability Assessment is provided to demonstrate ways in which an entity could violate the TSP and provide a list of identified vulnerabilities. Additionally, the document provides evidence of how the TOE is resistant to obvious attacks. The Strength of TOE Security Function Analysis demonstrates the strength of the probabilistic or permutational mechanisms employed to provide security functions within the TOE and how they exceed the minimum SOF requirements.

Corresponding CC Assurance Components:

- Strength of TOE Security Function analysis
- Vulnerability Analysis

8.5 Strength of Function

Strength of function rating of SOF-basic was claimed for this TOE to meet the EAL3+ assurance requirements, this SOF is sufficient to resist the threats identified in Section 3. Section 4 provides evidence that demonstrates that TOE threats are countered by the TOE security objectives. Section 8 demonstrates that the security objectives for the TOE and the TOE environment are satisfied by the security requirements. The evaluated TOE is intended to operate in commercial and DoD low robustness environments processing unclassified information.

The overall TOE SOF claim is SOF-basic because this SOF is sufficient to resist the threats identified in Section 3.1. Section 8.1 provides evidence that demonstrates that TOE threats are countered by the TOE security objectives. Section 8.2 demonstrates that the security objectives for the TOE and the TOE environment are satisfied by the security requirements.

The relevant security function and security functional requirement which has probabilistic or permutational functions is FIA_UAU.1.

9 Acronyms

Table 17: Acronyms

Acronym	Definition
ACID	Atomicity, Consistency, Isolation, And Durability
ACM	Access Control Matrix
CC	Common Criteria
CLI	Command Line Interface
CM	Configuration management
EAL	Evaluation Assurance Level
GB	Gigabyte
GUI	Graphical user interface
ISO	International Organization for Standardization
IT	Information technology
JDBC	Java Database Connectivity
ODBC	Open Database Connectivity
OS	Operating system
SAR	Security Assurance Requirement
SFP	Security Function Policy
SFR	Security Functional Requirement
SOF	Strength of function
SPU	Snippet Processing Unit
SQL	Structured Query Language
ST	Security Target
TOE	Target of Evaluation
TBA	To be announced
TBD	To be determined
TSF	Target of Evaluation (TOE) security function
TSP	Target of Evaluation (TOE) security policy