IBM
Informix Dynamic Server Version 11.5
Security Target

Version 1.0
September 25, 2008

Prepared for:

International Business Machines

11200 Lakeview Avenue
Lenexa, KS 66219

Prepared By:

Science Applications International Corporation

Common Criteria Testing Laboratory

7125 Columbia Gateway Drive, Suite 300
Columbia, MD 21046

LIST OF TABLES

1. Security Target Introduction

This section identifies the Security Target (ST) and Target of Evaluation (TOE) identification, ST conventions, ST conformance claims, and the ST organization. The TOE is IBM Informix Dynamic Server Version 11.5 (IDS) provided by International Business Machines. The IBM IDS product is a relational database management system (RDBMS) sold as an application to be installed on a commercial operating system.

The Security Target contains the following additional sections:

- TOE Description (Section 2)
- Security Environment (Section 3)
- Security Objectives (Section 4)
- IT Security Requirements (Section 5)
- TOE Summary Specification (Section 6)
- Protection Profile Claims (Section 7)
- Rationale (Section 8).

1.1 Security Target, TOE and CC Identification

ST Title – IBM Informix Dynamic Server Version 11.5 Security Target

ST Version – Version 1.0

ST Date – September 25, 2008

TOE Identification – IBM Informix Dynamic Server Version 11.5 (Enterprise Editions)

TOE Developer – IBM

Evaluation Sponsor – IBM

CC Identification – Common Criteria for Information Technology Security Evaluation, Version 2.3, August 2005

1.2 Conformance Claims

This TOE is conformant to the following CC specifications:

- Common Criteria for Information Technology Security Evaluation Part 2: Security Functional Requirements, Version 2.3, August 2005.
  - Part 2 Extended
- Common Criteria for Information Technology Security Evaluation Part 3: Security Assurance Requirements, Version 2.3, August 2005.
  - Part 3 Conformant
  - Assurance Level: EAL 4 augmented with ALC_FLR.2
  - Strength of Function Claim: SOF-medium

1.3 Conventions

The following conventions have been applied in this document:

- Security Functional Requirements – Part 2 of the CC defines the approved set of operations that may be applied to functional requirements: iteration, assignment, selection, and refinement.
  - Iteration: allows a component to be used more than once with varying operations. In the ST, iteration is indicated by a letter placed at the end of the component. For example FDP_ACC.1a and FDP_ACC.1b indicate that the ST includes two iterations of the FDP_ACC.1 requirement, a and b.
  - Assignment: allows the specification of an identified parameter. Assignments are indicated using bold and are surrounded by brackets (e.g., [**assignment**]). Note that an assignment within a selection would be identified in italics and with embedded bold brackets (e.g., [***[selected-assignment]***]).
  - Selection: allows the specification of one or more elements from a list. Selections are indicated using bold italics and are surrounded by brackets (e.g., [***selection***]).
  - Refinement: allows the addition of details. Refinements are indicated using bold, for additions, and strike-through, for deletions (e.g., "… **all** objects …" or "… ~~some~~ **big** things …").
- Explicitly stated requirements (i.e., those not found in the CC) are identified with '(explicitly stated requirement)' in its corresponding paragraph title.
- Other sections of the ST – Other sections of the ST use bolding to highlight text of special interest, such as captions.

2. TOE Description

The Target of Evaluation (TOE) is IBM Informix Dynamic Server Version 11.5 (IDS) Enterprise Editions.

The IBM IDS 11.5 is the current version of a prior release of the same product that had been previously evaluated. INFORMIX-Online/Secure 5.0, which has since been renamed Information Dynamic Server, was evaluated under the National Security Agency (NSA) Trusted product Evaluation Program (TPEP) at the C2 and B1 levels of the Trusted Database Interpretation (TDI) of the Trusted Computer System Evaluation Criteria (TCSEC) in 1994.

While the product has had undergone several changes, the claims in this Security Target are derived primarily from the Common Criteria (CC) equivalent of the C2 TCSEC requirements as embodied in the U.S. government Controlled Access Protection Profile (CAPP). However, since the CAPP is intended for use with operating system (OS) products, the claims have been altered to account for the differences between RDBMS and OS type products.

Note that while there are DBMS-specific PPs, conformance cannot be achieved due to certain requirements that could be met by few if any current DBMS products. Or particular note are the requirements dictating that the DBMS must be able to limit the number of concurrent connections a given user can have; the DBMS must store and retrieve date/time information associated with sessions, and the DBMS must be able to restrict sessions based user/group identity, time of day and day of week.

2.1 TOE Overview

The IDS is an RDBMS designed primarily to implement databases that can be manipulated using Structured Query Language (SQL) statements.

The IDS is an application realized by a collection of cooperating processes. As an application, IDS depends on the underlying operating system for its execution environment and communication services as well as for storage mechanisms for itself, its configuration, and its databases. It also depends on the underlying operating system for protection of its resources for its own protection and also for the differentiation and protection of its clients.

The IDS acts as a server servicing requests of local clients on the same host operating system and on other hosts using network communication mechanisms. The IDS offers a proprietary SQLI protocol to its own clients as well as Distributed Relational Database Architecture (DRDA) support for other clients.

2.2 TOE Architecture

The IDS is a multi-process and multi-threaded application. Each process of the IDS application is referred to as a Virtual Processor (VP) and each VP is designed to fulfill a specific role in implementing the RDBMS. There are VPs specifically designed to handle SQL statements, network communication, local communication, I/O processing, and other miscellaneous functions of IDS. Each of the processes of IDS share memory resources and file descriptors, working as a collective. The processing for a given session can move from VP to VP as necessary. This happens when threads in one VP call threads in another VP to continue a logical thread of execution for the session, utilizing resources (e.g., stack) stored in shared memory. Multiple threads can be used to achieved parallelism for a given session when appropriate (e.g., for parallel sorts and scans). Most of the actual SQL processing is accomplished on CPU VPs using non-preemptive scheduling for threads. When a thread goes into a wait state, the VP switches stacks and continues with another thread.

2.2.1 Physical Boundaries

The TOE is IBM Informix Dynamic Server Version 11.5. The main program for the IDS, used for all VPs, is ‘oninit’. The TOE includes a number of additional utility programs for the purposes of managing IDS. A complete list can be found in the administrator guidance documents, but the more security relevant utilities are:

- onmode: provides means to modify behavior and state of the engine; supports adding and dropping of VPs
- onspaces: dbspace (tablespace) and chunk (container) administration
- onparams: provides a means to dynamically add or drop logs
- onaudit: manages audit masks and auditing configuration
- onshowaudit: extracts information from an audit trail
- dbload: load data into a database table
- dbaccess: a client application distributed with the product that facilitates communication between database users (e.g., administrators) and the database VPs

Note that there are other products, including Informix Connect, Informix DataBlade Developer’s Kit, Informix Server Administrator (ISA) and Informix Spatial Datablade, associated with IDS (e.g., that may be referenced in guidance documents) that are not included within the TOE because they are separate products subject to separate license requirements.

The IDS is design to operate on a number of UNIX operating systems as well as Microsoft Windows as indicated below:

Version	Platform	Processor Model	OS Build
Sun 32-bit	Solaris	Sparc	Solaris 9, Solaris10
Sun 64-bit	Solaris	Sparc	Solaris 9, Solaris10
Sun 64-bit	Solaris	AMD64 (Opteron)	Solaris 10
HP 32-bit	HP-UX	PA-RISC	HP-UX 11i, HP-UX 11.23PI, 11.31
HP 64-bit	HP-UX	PA-RISC	HP-UX 11i, HP-UX 11.23PI,11.31
HP 64-bit	HP-UX	Itanium	HP-UX 11.23PI, HP-UX 11.31
HP 32-bit	HP-UX	Itanium	HP-UX 11.23PI, HP-UX11.31
IBM 32-bit	AIX	PowerPC	AIX 5L 5.3
IBM 64-bit	AIX	PowerPC	AIX 5L 5.3
Windows	Windows	x86	Windows 2003, Windows XP,Vista
Intel 32-bit	Linux	x86	RHEL 4, SUSE SLES 10, Asianux 2.05
Intel/AMD 32-bit	Linux	x86_64 (EM64T/AMD64)	RHEL 4, SUSE SLES 10, Asianux 2.05
Intel/AMD 64-bit	Linux	x86_64 (EM64T/AMD64)	RHEL 4, SUSE SLES 10, Asianux 2.05
IBM 64-bit	Linux	PowerPC	RHEL 4, SUSE SLES 10, Asianux 2.05

Version	Platform	Processor Model	OS Build
		(pSeries/iSeries, OpenPower, JS20 Blades)	
IBM 64-bit	Linux	zSeries	RHEL 4, SUSE SLES 10
Intel 64-bit	Linux	Itanium	RHEL 4, SUSE SLES 10
Solaris Opteron 32 bit client only	Solaris	Opteron	

Table 1 Supported PltformsPlatforms

Additionally, IDS can be configured to use a pluggable authentication module (PAM) implemented within the IT environment in order to ensure that users are authenticated properly. This is an alternative to relying on authentication that otherwise would be provided by the underlying operating system.

2.2.2 Logical Boundaries

The logical boundaries of IDS are realized in the security functions that it implements. These security functions are realized at the IDS interfaces that service client requests (SQLI and DRDA for both local and remote clients) and via the administrator commands identified above.

2.2.2.1 Security Audit

The IDS has the ability to audit security relevant events related to its security functions. An authorized administrator, using the *onaudit* utility program, can enable and disable the audit feature and can select specifically which security relevant events should be audited based on event type and user.

Audit records are stored within files in the IT environment. The *onshowaudit* utility allows an authorized administrator to extract the audit records from the audit trail into a file that could potentially be viewed directly using tools available in the IT environment or alternately it can be loaded into an IDS database table, using *dbload*, so that the features of IDS can be used to more effectively review the audit records with searching and sorting capabilities.

2.2.2.2 Access Control

The IDS associates privileges with each individual user. These privileges are associated with operations that can be performed on the objects (e.g., database) that are implemented by the IDS. The IDS uses identities, privileges, and access control lists associated with users and objects to determine whether specific operations will be allowed when attempted by client users.

IDS implements a few roles, each having special privileges that are not available to normal users. These roles are associated with groups defined in the underlying operating system and users are assigned roles by virtue of their membership in those groups. Note that users in these roles can execute certain privileged SQL commands while ‘privileges’ are associated with access permissions for IDS objects. For this ST, references to the “authorized administrator” role are implemented in the IDS as any of the following roles: Operating System Administrator (OSA), Database System Security Officer (DBSSO), Database System Administrator (DBSA), Database Security Administrator (DBSECADM), or Audit Analysis Officer (AAO). While the IDS offers these different roles with distinct responsibilities, this ST does not make specific role separation claims and hence treats them all logically as a single role – the authorized administrator. References to the “user” role are implemented in the IDS as any user not a member of one of the administrative roles.

Note that by default, user *informix* is the DBSA and group *informix* is the DBSA group. This user and group is used at server installation as the owner and group of the IDS installation and as such provides the OS level protection of who can access and use files in the IDS installation. In particular, the directory defined as $INFORMIXDIR (the IDS server base installation directory) must be owned by user *informix* and group *informix*.

In addition to using privileges and authorities to control access, IDS implements a label-based access control (LBAC) mechanism. The IDS DBSECADM can grant (or revoke) security labels and exemptions to (or from) users as well as create and drop LBAC security objects in order to define LBAC polices for specific database tables. Once a table is configured with a LBAC policy (i.e., the table is LBAC protected relative to either rows or columns), users must additionally satisfy the LBAC access rules in order to access or modify the applicable table rows or columns.

Note also that the IDS is designed to carefully manage it resources to ensure that information is not inadvertently shared when a database object is created or otherwise results in the reuse of underlying IDS resources. While the IDS manages its own resources to this end, it relies on its environment to appropriately clear or initialize its resources that serve to instantiate the resources of IDS.

2.2.2.3 Identification & Authentication

The IDS requires all users to be identified before allowing them access to IDS resources. The IT environment is responsible for user authentication while the IDS requires the user identity returned by the IT environment to associate IDS credentials (e.g., privileges) with the authenticated user.

2.2.2.4 Security Management

The IDS includes the roles of authorized administrator and user implemented using IT environment groups, and associated IDS roles (see above) and (access control) privileges, and allows individual users to be assigned to those roles by virtue of the assignment of the applicable groups (in the IT environment) and privileges to their identity. Management of the IDS TOE, including the ability to select and review audit records, is restricted to authorized administrators and access to the TOE (e.g., the utility programs and associated data and configuration files) through its IT environment. Management of the IDS objects is restricted to those users that are assigned the appropriate privileges to do so.

Note that for the most part management of the TOE is accomplished via SQL statements that can be issued interactively using the *dbaccess* utility.

2.2.2.5 TOE Protection

The IDS executes within processes provided by the host operating system. However, it is designed to not share its process space with non-TOE entities in order to ensure that its resources are protected. The IDS has been designed so that each of its interfaces performs the necessary access checks before allowing access to IDS resources.

2.3 TOE Documentation

There is an extensive set of user and administrator guidance documents available for the IDS product. See section 6.2 for specific details.

3. Security Environment

Since IDS was developed with consideration of the Trusted Computer System Evaluation Criteria (TCSEC) C2 security requirements, the security environment has been modeled after that specified in the Controlled Access Protection Profile (CAPP), which is the successor to TCSEC C2 in the context of the Common Criteria (CC). Note, however, that since IDS is a database system and not an operating system, some additional assumptions and security objectives have been assigned to the IT environment of the TOE.

3.1 Secure Usage Assumptions

The usage assumptions are organized in three categories: personnel (assumptions about administrators and users of the system as well as any threat agents), physical (assumptions about the physical location of the TOE or any attached peripheral devices), and connectivity (assumptions about other IT systems that are necessary for the secure operation of the TOE).

3.1.1 Personnel Assumptions

It is assumed that the following personnel conditions will exist:

A.MANAGE

There will be one or more competent individuals assigned to manage the TOE and the security of the information it contains.

A.NO_EVIL_ADM

The system administrative personnel are not careless, willfully negligent, or hostile, and will follow and abide by the instructions provided by the administrator documentation.

A.COOP

Authorized users possess the necessary authorization to access at least some of the information managed by the TOE and are expected to act in a cooperating manner in a benign environment.

A.CLEARANCE

Procedures exist for granting users authorization for access to specific security levels. It is further assumed the TOE administrators will be cleared to the highest security level processed by the TOE.

3.1.2 Physical Assumptions

The TOE is intended for application in areas that have physical control and monitoring. It is assumed that the following physical conditions will exist:

A.LOCATE

The processing resources of the TOE will be located within controlled access facilities, which will prevent unauthorized physical access.

A.PROTECT

The hardware and software critical to security policy enforcement will be protected from unauthorized physical modification.

3.1.3 Connectivity Assumptions

It is assumed that the following connectivity conditions exist:

A.CONNECT

All connections to peripheral devices reside within the controlled access facilities. The TOE only addresses security concerns related to the manipulation of the TOE through its authorized access points. Internal communication paths to access points such as terminals are assumed to be adequately protected.

A.PLATFORM

The IT Environment underlying the TOE is assumed to fulfill the requirements for the IT Environment described in this ST. It is also assumed that the IT Environment will provide a suitable operational environment for the TOE where the TOE will be able to properly execute and the dependencies that the TOE has upon the IT Environment are properly fulfilled.

3.2 Threats

All security objectives, except for the non-IT security objectives for the environment, have been derived from the statement of Organizational Security Policy found in the following section. Non-IT security objectives for the environment have been drawn from the Secure Usage Assumptions detailed in Section 3.1. Therefore, there is no statement of the explicit threats countered by the TOE.

3.3 Organization Security Policies

An Organizational Security Policy is a set of rules or procedures imposed by an organization upon its operations to protect its sensitive data. Although some of the organizational security policies described below are drawn from the CAPP they apply to many non-DoD environments.

P.AUTHORIZED_USERS

Only those users who have been authorized to access the information within the TOE may access the TOE.

P.NEED_TO_KNOW

The TOE must limit the access to, modification of, and destruction of the information in protected resources to those authorized users which have a “need to know” for that information.

P.ACCOUNTABILITY

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

P.CLASSIFICATION

The system must be able to limit the access to information based on sensitivity, as represented by a label, of the information contained in objects, and the formal clearance of users, as represented by subjects, to access that information. The access rules enforced prevent a subject from accessing information which is of higher sensitivity than it is operating at.

4. Security Objectives

This section defines the security objectives of the TSF and its supporting environment. Security objectives, categorized as either applying to the TOE or its environment, reflect the stated intent to comply with any assumptions and organizational security policies identified. All of the identified assumptions and organizational policies are addressed under one of the categories below.

4.1 Security Objectives for the TOE

O.AUTHORIZATION

The TSF must ensure that only authorized users gain access to the TOE and its resources.

O.DISCRETIONARY_ACCESS

The TSF must control accesse to resources based on identity of users. The TSF must allow authorized users to specify which users may access which resources.

O.MANDATORY_ACCESS

The TSF must be able to control access to resources based upon the sensitivity and categories of the information being accessed and the clearance of the subject attempting to access that information.

O.AUDITING

The TSF must record the security relevant actions of users of the TOE. The TSF must present this information to authorized administrators.

O.RESIDUAL_INFORMATION

The TSF must ensure that any information contained in a protected resource is not released when the resource is recycled.

O.MANAGE

The TSF must provide all the functions and facilities necessary to support the authorized administrators that are responsible for the management of TOE security.

O.ENFORCEMENT

The TSF must be designed and implemented in a manner that ensures that the organizational policies are enforced in the target environment.

4.2 Security Objectives for the Environment

The TOE is assumed to be complete and self-contained and, as such, is not dependent upon any other products to perform properly. However, certain objectives with respect to the general operating environment must be met. The following are the security objectives for the environment:

4.2.1 Non-IT security objectives for the environment

O.ADMIN_GUIDANCE

Appropriate guidance documentation must be provided to enable administrators to install, manage, and operate the TOE in a manner that maintains IT security objectives.

O.ADMINISTRATORS

Administrators of the TOE and IT Environment must not be careless, willfully negligent or hostile, and must follow the instructions provided in the administrator guidance documentation.

O.ASSIGN

One or more competent individuals must be assigned to manage the TOE and the security of the information it contains.

O.COOP

Authorized users must possess the appropriate authorization to access at least some of the information managed by the TOE and must act in a cooperative manner in a benign environment.

O.INSTALL

Those responsible for the TOE must ensure that the TOE is delivered, installed, managed, and operated in a manner, which maintains IT security objectives.

O.PHYSICAL

Those responsible for the TOE must ensure that those parts of the TOE critical to security policy are protected from physical attack, which might compromise IT security objectives.

O.CREDEN

Those responsible for the TOE must ensure that all access credentials, such as passwords or other authentication information, are protected by the users in a manner that maintains IT security objectives and that credentials (e.g., clearances) are assigned appropriately.

O.PLATFORM

The IT Environment underlying the TOE must fulfill the requirements for the IT Environment described in this ST. The IT Environment must provide a suitable operational environment for the TOE where the TOE is able to properly execute and the dependencies that the TOE has upon the IT Environment are properly fulfilled

4.2.2 IT security objectives for the environment

OE.AUTHORIZATION

The IT Environment must ensure that only authorized users gain access to the IT Environment and its resources. The IT Environment must support the TOE by ensuring that users are adequately authenticated on the TOE’s behalf.

OE.AUDITING

The IT Environment must record the security relevant actions of users of the IT Environment.

OE.RESIDUAL_INFORMATION

The IT Environment must ensure that any information contained in a protected resource is not released when the resource is recycled.

OE.MANAGE

The IT Environment must provide all the functions and facilities necessary to support the authorized administrators that are responsible for the management of IT Environment security, including security relevant support for the TOE.

OE.ENFORCEMENT

The IT Environment must be designed and implemented in a manner that ensures that it can protect the operational IT Environment of the TOE. The IT Environment must provide a reliable time source for the use of both the TOE and the IT Environment.

5. IT Security Requirements

The following sections define the security functional and assurance requirements for the TOE and its IT environment. The security functional requirements have been drawn largely from the Controlled Access Protection Profile (CAPP) and the security assurance requirements have been drawn from EAL 4, as defined in the CC Part 3, augmented with ALC_FLR.2.

Note that this ST includes the security assurance requirement AVA_SOF.1, requiring strength of function analysis to demonstrate that each probabilistic or permutational security mechanism meets the stated strength of function (SOF) claim. However, the TOE does not include any probabilistic or permutational security mechanisms and, as a result, while this ST makes a minimum SOF claim of SOF-medium, it is not really applicable.

5.1 TOE Security Functional Requirements

This section specifies the security functional requirements that are applicable to the TOE.

Security Functional Class	Security Functional Components
Security Audit (FAU)	FAU_GEN.1a Audit data generation
	FAU_GEN.2 User identity association
	FAU_SAR.1 Audit review
	FAU_SAR.2 Restricted audit review
	FAU_SAR.3 Selectable audit review
	FAU_SEL.1 Selective audit
	FAU_STG.4 Prevention of audit data loss
User Data Protection (FDP)	FDP_ACC.1 Subset access control
	FDP_ACF.1 Security attribute based access control
	FDP_IFC.1 Subset information flow control
	FDP_IFF.2 Hierarchical security attributes
	FDP_RIP.2a Full residual information protection
Identification and authentication (FIA)	FIA_ATD.1a User attribute definition
	FIA_UID.2a User identification before any action
	FIA_USB.1 User-subject binding
Security management (FMT)	FMT_MOF.1 Management of security functions behaviour
	FMT_MSA.1a Management of Security Attributes
	FMT_MSA.1b Management of Security Attributes
	FMT_MSA.3a Static Attribute Initialization
	FMT_MSA.3b Static Attribute Initialization
	FMT_REV.1a Revocation
	FMT_SMF.1a Specification of Management Functions
	FMT_SMR.1a Security roles
Protection of the TSF (FPT)	FPT_RVM.1a Non-bypassability of the TSP
	FPT_STM.1a Reliable Time Stamps **(explicitly stated)**

Table 2 TOE Functional Security Requirements

5.1.1 Security Audit (FAU)

5.1.1.1 Audit data generation (FAU_GEN.1a)

FAU_GEN.1a.1 The TSF shall be able to generate an audit record of the following auditable events:

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

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

c) [**object access attempts, use of privileged SQL statements, changes in identity, changing the logging mode, and user connection attempts**].

FAU_GEN.1a.2 The TSF shall record within each audit record at least the following information:

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

b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [**no additional details**].

5.1.1.2 User identity association (FAU_GEN.2)

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

5.1.1.3 Audit review (FAU_SAR.1)

FAU_SAR.1.1 The TSF shall provide [**authorised 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.

5.1.1.4 Restricted audit review (FAU_SAR.2)

FAU_SAR.2.1 The TSF shall prohibit all users read access to the audit records, except those users that have been granted explicit read-access.

5.1.1.5 Selectable audit review (FAU_SAR.3)

FAU_SAR.3.1 The TSF shall provide the ability to perform [***sorting***, ***searches***] of audit data based on [**user identity and event type**].

5.1.1.6 Selective audit (FAU_SEL.1)

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

a) [***event type, user identity***]

b) [**no additional attributes**].

5.1.1.7 Prevention of audit data loss (FAU_STG.4)

FAU_STG.4.1 The TSF shall [***'ignore auditable events'***] **or 'prevent auditable events' as configured an authorized administrator** and [**no other action**] if the audit trail is full.

5.1.2 User Data Protection (FDP)

5.1.2.1 Subset access control (FDP_ACC.1)

FDP_ACC.1.1 The TSF shall enforce the [**Discretionary Access Control Policy**] on [**user attempts to create, destroy or otherwise access databases, tables, views, synonyms, types, routines, and sequences**].

5.1.2.2 Security attribute based access control (FDP_ACF.1)

FDP_ACF.1.1 The TSF shall enforce the [**Discretionary Access Control Policy**] to objects based on the following: [**subject and object attributes as defined in the table below**].

Controlled entity	Security attributes
Subjects	
User	Username and roles
Objects	
Database	Access control list[1]
Table View Synonym Type Routine Sequence	Access control list Owner

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [**a subject must have a username that is assigned the privilege (per the access control list) corresponding to the requested operation of the target object in order to succeed in performing the requested operation**].

FDP_ACF.1.3 The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: [1) **a subject that is an authorized administrator can access objects as allowed by their roles regardless of privileges (in the access control list) and 2) a subject that has a username that is the owner of the applicable object can access the object regardless of privileges**].

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

5.1.2.3 Subset information flow control (FDP_IFC. 1)

FDP_IFC.1.1 The TSF shall enforce the [**LBAC SFP**] on [**user read and write operations on LBAC protected database tables**].

5.1.2.4 Hierarchical security attributes (FDP_IFF.2)

FDP_IFF.2.1 The TSF shall enforce the [**LBAC Policy**] based on the following types of subject and information security attributes: [**user security labels and database table column or row security labels**].

Application Note:
Note that security labels consist of zero (0) or more of each of the three (3) available component types (array, set, and tree), but must include at least one component.

- Array – represents an ordered set; any element in the set is ranked higher than subsequent elements in the set.
- Set – represents an unordered set; there is no defined relationship among the elements in the set and there order is not important.
- Tree – represents a hierarchy and is used to represent organizational charts and to identify departments within an organization that owns the applicable data. An element of a tree that is higher than another element in the tree hierarchy is considered an *ancestor*.

[1] Access control lists assign privileges to users via Usernames. Note that privileges in this context can be viewed as access permissions. The more traditional notion of privileges is represented by roles that can access privileged SQL statements in this ST.

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

1) **in order to read a LBAC protected column or row in a database table:**

a) the array components of the user's security label must be greater than or equal to the array components of the object's security label,

b) the set components of the user's security label must include the set components of the object's security label, and

c) the tree components of the user's security label must include at least one of the elements in the tree components of the object's security label (or the ancestor of one such element) and

2) in order to write a LBAC protected column or row in a database table:

a) the array components of the user's security label must be equal to the array components of the object's security label,

b) the set components of the user's security label must include the set components of the object's security label, and

c) the tree components of the user's security label must include at least one of the elements in the tree components of the object's security label (or the ancestor of one such element) and

3) the Discretionary Access Control Policy rules must be satisfied in every case].

FDP_IFF.2.3 The TSF shall enforce the [**no additional rules**].

FDP_IFF.2.4 The TSF shall provide the following [**only a security administrator can change security labels on users and an appropriately privileged user can change security labels on columns or rows of LBAC protected tables**].

FDP_IFF.2.5 The TSF shall explicitly authorise an information flow based on the following rules: [a **user with the appropriate corresponding exemption can ignore the read array, read set, read tree, write array (to lower array values), write array (to higher array values), write set, or write tree check**].

FDP_IFF.2.6 The TSF shall explicitly deny an information flow based on the following rules: [**none**].

FDP_IFF.2.7 The TSF shall enforce the following relationships for any two valid information flow control security attributes: a) There exists an ordering function that, given two valid security attributes, determines if the security attributes are equal, if one security attribute is greater than the other, or if the security attributes are incomparable; and b) There exists a 'least upper bound' in the set of security attributes, such that, given any two valid security attributes, there is a valid security attribute that is greater than or equal to the two valid security attributes; and c) There exists a 'greatest lower bound' in the set of security attributes, such that, given any two valid security attributes, there is a valid security attribute that is not greater than the two valid security attributes.

5.1.2.5 Full residual information protection (FDP_RIP.2a)

FDP_RIP.2a.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon the [***allocation of the resource to***] all objects.

5.1.3 Identification and authentication (FIA)

5.1.3.1 User attribute definition (FIA_ATD.1a)

FIA_ATD.1a.1 The TSF shall maintain the following list of security attributes belonging to individual users: [**username, roles, LBAC security label, and LBAC exemptions**].

5.1.3.2 User identification before any action (FIA_UID.2a)

FIA_UID.2a.1 The TSF shall require each user to identify itself before allowing any other TSF-mediated actions on behalf of that user.

5.1.3.3 User-subject binding (FIA_USB.1)

FIA_USB.1.1 The TSF shall associate the following user security attributes with subjects acting on the behalf of that user: [**username and roles**].

FIA_USB.1.2 The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users: [**when user session is created, the username is obtained from the host operating system and roles are associated with the session per OS groups associated with the username**].

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: [**once a session is created its attributes do not change, except when changed by an authorized user**].

5.1.4 Security management (FMT)

5.1.4.1 Management of security functions behaviour (FMT_MOF.1)

FMT_MOF.1.1 The TSF shall restrict the ability to [***modify the behaviour of***] the functions [**LBAC Policy**] to [**an authorized administrator**].

5.1.4.2 Management of Security Attributes (FMT_MSA.1a)

FMT_MSA.1a.1 The TSF shall enforce the [**Discretionary Access Control Policy**] to restrict the ability to [***modify***] the security attributes [**access control list and owner**] to [**users authorized by the Discretionary Access Control Rules**].

5.1.4.3 Management of Security Attributes (FMT_MSA.1b)

FMT_MSA.1b.1 The TSF shall enforce the [**LBAC Policy**] to restrict the ability to [***modify***] the security attributes [**database table column or row security labels**] to [**users authorized by the LBAC Rules**].

5.1.4.4 Static Attribute Initialization (FMT_MSA.3a)

FMT_MSA.3a.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~~ **Discretionary Access Control Policy**.

FMT_MSA.3a.2 The TSF shall allow the [**no role**] to specify alternative initial values to override the default values when an object or information is created.

5.1.4.5 Static Attribute Initialization (FMT_MSA.3b)

FMT_MSA.3b.1 The TSF shall enforce the [**LBAC Policy**] to provide [***[no]***] default values for security attributes that are used to enforce the ~~SFP~~ **LBAC Policy**.

FMT_MSA.3b.2 The TSF shall allow the [**no role**] to specify alternative initial values to override the default values when an object or information is created.

5.1.4.6 Revocation (FMT_REV.1a)

FMT_REV.1a.1 The TSF shall restrict the ability to revoke security attributes associated with the [***objects***] within the TSC to [**users authorised to modify the Discretionary Access Control security attributes by the Discretionary Access Control policy**].

FMT_REV.1a.2 The TSF shall enforce the rules [**the access rights associated with an object shall be enforced when an access check is made**].

5.1.4.7 Specification of Management Functions (FMT_SMF.1a)

FMT_SMF.1a.1 The TSF shall be capable of performing the following security management functions: [**start and stop auditing; select audited events; review the audit trail; management of Discretionary Access Control privileges; create, rename, and drop LBAC policies and labels; and, grant and revoke LBAC security labels and exemptions**].

5.1.4.8 Security Roles (FMT_SMR.1a)

FMT_SMR.1a.1 The TSF shall maintain the roles [**authorised administrator and user**].

FMT_SMR.1a.2 The TSF shall be able to associate users with roles.

5.1.5 Protection of the TSF (FPT)

5.1.5.1 Non-bypassability of the TSP (FPT_RVM.1a)

FPT_RVM.1a.1 The TSF shall ensure that TSP enforcement functions are invoked and succeed before each function within the TSC is allowed to proceed.

5.1.5.2 Reliable Time Stamps (FPT_STM.1a) (explicitly stated requirement)

FPT_STM.1a.1 The TSF shall be able to provide reliable time stamps based on information provided by the IT environment for its own use.

5.2 Security Requirements for the IT Environment

This section specifies the security requirements that are applicable to IT environment of the TOE.

Security Functional Class	Security Functional Components
Security Audit (FAU)	FAU_GEN.1b Audit data generation
	FAU_STG.1: Guarantees of Audit Data Availability
User Data Protection (FDP)	FDP_RIP.2b Full residual information protection
Identification and authentication (FIA)	FIA_ATD.1b User attribute definition
	FIA_SOS.1 Verification of secrets
	FIA_UAU.2 User authentication before any action
	FIA_UAU.7 Protected authentication feedback
	FIA_UID.2b User identification before any action
Security management (FMT)	FMT_MTD.1a Management of TSF data
	FMT_MTD.1b Management of TSF data
	FMT_MTD.1c Management of TSF data
	FMT_MTD.1d Management of TSF data
	FMT_MTD.1e Management of TSF data
	FMT_REV.1b Revocation
	FMT_SMF.1b Specification of Management Functions
	FMT_SMR.1b Security Management Roles
Protection of the TSF (FPT)	FPT_AMT.1 Abstract Machine Testing
	FPT_RVM.1b Reference Mediation
	FPT_SEP.1 Domain Separation
	FPT_STM.1b Reliable Time Stamps

Table 3 IT Environment Functional Security Requirements

5.2.1 Security Audit (FAU)

5.2.1.1 Audit data generation (FAU_GEN.1b)

FAU_GEN.1b.1 The ~~TSF~~ **IT Environment** shall be able to generate an audit record of the following auditable events:

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

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

c) [**attempts to log in (to the host operating system) and use of security management functions**].

FAU_GEN.1b.2 The ~~TSF~~ **IT Environment** shall record within each audit record at least the following information:

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

b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [**no additional details**].

5.2.1.2 Guarantees of Audit Data Availability (FAU_STG.1)

FAU_STG.1.1 The ~~TSF~~ **IT Environment** shall protect the stored audit records from unauthorized deletion.

FAU_STG.1.2 The ~~TSF~~ **IT Environment** shall be able to prevent unauthorised modifications to the audit records in the audit trail.

5.2.2 User Data Protection (FDP)

5.2.2.1 Full residual information protection (FDP_RIP.2b)

FDP_RIP.2b.1 The ~~TSF~~ **IT Environment** shall ensure that any previous information content of a resource is made unavailable upon the [***allocation of the resource to***] all objects.

5.2.3 Identification and authentication (FIA)

5.2.3.1 User attribute definition (FIA_ATD.1b)

FIA_ATD.1b.1 The ~~TSF~~ **IT Environment** shall maintain the following list of security attributes belonging to individual users: [**user identifier, group memberships, roles, and authentication data**].

5.2.3.2 Verification of secrets (FIA_SOS.1)

FIA_SOS.1.1 The ~~TSF~~ **IT Environment** shall provide a mechanism to verify that secrets meet [**the following**

a) for each attempt to use the authentication mechanism, the probability that a random attempt will succeed is less than one in 1,000,000;

b) for multiple attempts to use the authentication mechanism during a one minute period, the probability that a random attempt during that minute will succeed is less than one in 100,000; and

c) any feedback given during an attempt to use the authentication mechanism will not reduce the probability below the above metrics].

5.2.3.3 User authentication before any action (FIA_UAU.2)

FIA_UAU.2.1 The ~~TSF~~ **IT Environment** shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

5.2.3.4 Protected authentication feedback (FIA_UAU.7)

FIA_UAU.7.1 The ~~TSF~~ **IT Environment** shall provide only [**obscured feedback**] to the user while the authentication is in progress.

5.2.3.5 User identification before any action (FIA_UID.2b)

FIA_UID.2b.1 The ~~TSF~~ **IT Environment** shall require each user to identify itself before allowing any other TSF-mediated actions on behalf of that user.

5.2.4 Security management (FMT)

5.2.4.1 Management of TSF data (FMT_MTD.1a)

FMT_MTD.1a.1 The ~~TSF~~ **IT Environment** shall restrict the ability to [***delete* and *[create]***] the [**audit trail**] to [**authorised administrators**].

5.2.4.2 Management of TSF data (FMT_MTD.1b)

FMT_MTD.1b.1 The ~~TSF~~ **IT Environment** shall restrict the ability to [***modify* and *[observe]***] the [**set of audited events**] to [**authorised administrators**].

5.2.4.3 Management of TSF data (FMT_MTD.1c)

FMT_MTD.1c.1 The ~~TSF~~ **IT Environment** shall restrict the ability to [***modify* and *[initialize]***] the [**user security attributes other than authentication data**] to [**authorised administrators**].

5.2.4.4 Management of TSF data (FMT_MTD.1d)

FMT_MTD.1d.1 The ~~TSF~~ **IT Environment** shall restrict the ability to [***[initialize]***] the [**authentication data**] to [**authorised administrators].**

5.2.4.5 Management of TSF data (FMT_MTD.1e)

FMT_MTD.1e.1 The ~~TSF~~ **IT Environment** shall restrict the ability to [***modify***] the [**authentication data**] to [**the following: authorised administrators and users authorised to modify their own authentication data**].

5.2.4.6 Revocation (FMT_REV.1b)

FMT_REV.1b.1 The ~~TSF~~ **IT Environment** shall restrict the ability to revoke security attributes associated with the [***users***] within the TSC to [**authorised administrators**].

FMT_REV.1b.2 The ~~TSF~~ **IT Environment** shall enforce the rules [**the attributes associated with users shall be applied when a user is identified and authenticated**].

5.2.4.7 Specification of Management Functions (FMT_SMF.1b)

FMT_SMF.1b.1 The ~~TSF~~ **IT Environment** shall be capable of performing the following security management functions: [**create, modify, and delete user accounts**].

5.2.4.8 Security Management Roles (FMT_SMR.1b)

FMT_SMR.1b.1 The ~~TSF~~ **IT Environment** shall maintain the roles [**authorised administrator**].

FMT_SMR.1b.2 The ~~TSF~~ **IT Environment** shall be able to associate users with roles.

5.2.5 Protection of the TSF (FPT)

5.2.5.1 Abstract Machine Testing (FPT_AMT.1)

FPT_AMT.1.1 The ~~TSF~~ **IT Environment** shall run a suite of tests [***at the request of an authorised user***] to demonstrate the correct operation of the security assumptions provided by the abstract machine that underline the TSF.

5.2.5.2 Reference Mediation (FPT_RVM.1b)

FPT_RVM.1b.1 The ~~TSF~~ **IT Environment** shall ensure the TSP enforcement functions are invoked and succeed before each function within the TSC is allowed to proceed.

5.2.5.3 Domain Separation (FPT_SEP.1)

FPT_SEP.1.1 The ~~TSF~~ **IT Environment** 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~~ **IT Environment** shall enforce separation between the security domains of subjects in the TSC.

5.2.5.4 Reliable Time Stamps (FPT_STM.1b)

FPT_STM.1b.1 The ~~TSF~~ **IT Environment** shall be able to provide reliable time stamps for its own use **and for use by its subjects**.

5.3 TOE Security Assurance Requirements

The security assurance requirements for the TOE are the EAL 4 augmented with ALC_FLR.2 components as specified in Part 3 of the Common Criteria. No operations are applied to the assurance components.

Security Assurance Class	Security Assurance Components
ACM: Configuration management	ACM_AUT.1: Partial CM automation
	ACM_CAP.4: Generation support and acceptance procedures
	ACM_SCP.2: Problem tracking CM coverage
ADO: Delivery and operation	ADO_DEL.2: Detection of modification
	ADO_IGS.1: Installation, generation, and start-up procedures
ADV: Development	ADV_FSP.2: Fully defined external interfaces
	ADV_HLD.2: Security enforcing high-level design
	ADV_IMP.1: Subset of the implementation of the TSF
	ADV_LLD.1: Descriptive low-level design
	ADV_RCR.1: Informal correspondence demonstration
	ADV_SPM.1: Informal TOE security policy model
AGD: Guidance documents	AGD_ADM.1: Administrator guidance
	AGD_USR.1: User guidance
ALC: Life cycle support	ALC_DVS.1: Identification of security measures
	ALC_FLR.2: Flaw reporting procedures
	ALC_LCD.1: Developer defined life-cycle model
	ALC_TAT.1: Well-defined development tools
ATE: 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
AVA: Vulnerability assessment	AVA_MSU.2: Validation of analysis
	AVA_SOF.1: Strength of TOE security function evaluation
	AVA_VLA.2: Independent vulnerability analysis

Table 4 EAL 4 augmented with ALC_FLR.2 Assurance Components

5.3.1 Configuration management (ACM)

5.3.1.1 Partial CM automation (ACM_AUT.1)

ACM_AUT.1.1d The developer shall use a CM system.

ACM_AUT.1.2d The developer shall provide a CM plan.

ACM_AUT.1.1c The CM system shall provide an automated means by which only authorised changes are made to the TOE implementation representation.

ACM_AUT.1.2c The CM system shall provide an automated means to support the generation of the TOE.

ACM_AUT.1.3c The CM plan shall describe the automated tools used in the CM system.

ACM_AUT.1.4c The CM plan shall describe how the automated tools are used in the CM system.

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

5.3.1.2 Generation support and acceptance procedures (ACM_CAP.4)

ACM_CAP.4.1d The developer shall provide a reference for the TOE.

ACM_CAP.4.2d The developer shall use a CM system.

ACM_CAP.4.3d The developer shall provide CM documentation.

ACM_CAP.4.1c The reference for the TOE shall be unique to each version of the TOE.

ACM_CAP.4.2c The TOE shall be labelled with its reference.

ACM_CAP.4.3c The CM documentation shall include a configuration list, a CM plan, and an acceptance plan.

ACM_CAP.4.4c The configuration list shall uniquely identify all configuration items that comprise the TOE.

ACM_CAP.4.5c The configuration list shall describe the configuration items that comprise the TOE.

ACM_CAP.4.6c The CM documentation shall describe the method used to uniquely identify the configuration items that comprise the TOE.

ACM_CAP.4.7c The CM system shall uniquely identify all configuration items that comprise the TOE.

ACM_CAP.4.8c The CM plan shall describe how the CM system is used.

ACM_CAP.4.9c The evidence shall demonstrate that the CM system is operating in accordance with the CM plan.

ACM_CAP.4.10c The CM documentation shall provide evidence that all configuration items have been and are being effectively maintained under the CM system.

ACM_CAP.4.11c The CM system shall provide measures such that only authorised changes are made to the configuration items.

ACM_CAP.4.12c The CM system shall support the generation of the TOE.

ACM_CAP.4.13c The acceptance plan shall describe the procedures used to accept modified or newly created configuration items as part of the TOE.

ACM_CAP.4.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.1.3 Problem tracking CM coverage (ACM_SCP.2)

ACM_SCP.2.1d The developer shall provide a list of configuration items for the TOE.

ACM_SCP.2.1c The list of configuration items shall include the following: implementation representation; security flaws; and the evaluation evidence required by the assurance components in the ST.

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

5.3.2 Delivery and operation (ADO)

5.3.2.1 Detection of modification (ADO_DEL.2)

ADO_DEL.2.1d The developer shall document procedures for delivery of the TOE or parts of it to the user.

ADO_DEL.2.2d The developer shall use the delivery procedures.

ADO_DEL.2.1c The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to a user's site.

ADO_DEL.2.2c The delivery documentation shall describe how the various procedures and technical measures provide for the detection of modifications, or any discrepancy between the developer's master copy and the version received at the user site.

ADO_DEL.2.3c The delivery documentation shall describe how the various procedures allow detection of attempts to masquerade as the developer, even in cases in which the developer has sent nothing to the user's site.

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

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

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

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

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

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

5.3.3 Development (ADV)

5.3.3.1 Fully defined external interfaces (ADV_FSP.2)

ADV_FSP.2.1d The developer shall provide a functional specification.
ADV_FSP.2.1c The functional specification shall describe the TSF and its external interfaces using an informal style.
ADV_FSP.2.2c The functional specification shall be internally consistent.
ADV_FSP.2.3c The functional specification shall describe the purpose and method of use of all external TSF interfaces, providing complete details of all effects, exceptions and error messages.
ADV_FSP.2.4c The functional specification shall completely represent the TSF.
ADV_FSP.2.5c The functional specification shall include rationale that the TSF is completely represented.
ADV_FSP.2.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADV_FSP.2.2e The evaluator shall determine that the functional specification is an accurate and complete instantiation of the TOE security functional requirements.

5.3.3.2 Security enforcing high-level design (ADV_HLD.2)

ADV_HLD.2.1d The developer shall provide the high-level design of the TSF.
ADV_HLD.2.1c The presentation of the high-level design shall be informal.
ADV_HLD.2.2c The high-level design shall be internally consistent.
ADV_HLD.2.3c The high-level design shall describe the structure of the TSF in terms of subsystems.
ADV_HLD.2.4c The high-level design shall describe the security functionality provided by each subsystem of the TSF.
ADV_HLD.2.5c The high-level design shall identify any underlying hardware, firmware, and/or software required by the TSF with a presentation of the functions provided by the supporting protection mechanisms implemented in that hardware, firmware, or software.
ADV_HLD.2.6c The high-level design shall identify all interfaces to the subsystems of the TSF.
ADV_HLD.2.7c The high-level design shall identify which of the interfaces to the subsystems of the TSF are externally visible.
ADV_HLD.2.8c The high-level design shall describe the purpose and method of use of all interfaces to the subsystems of the TSF, providing details of effects, exceptions and error messages, as appropriate.
ADV_HLD.2.9c The high-level design shall describe the separation of the TOE into TSP-enforcing and other subsystems.
ADV_HLD.2.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADV_HLD.2.2e The evaluator shall determine that the high-level design is an accurate and complete instantiation of the TOE security functional requirements.

5.3.3.3 Subset of the implementation of the TSF (ADV_IMP.1)

ADV_IMP.1.1d The developer shall provide the implementation representation for a selected subset of the TSF.
ADV_IMP.1.1c The implementation representation shall unambiguously define the TSF to a level of detail such that the TSF can be generated without further design decisions.
ADV_IMP.1.2c The implementation representation shall be internally consistent.
ADV_IMP.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADV_IMP.1.2e The evaluator shall determine that the least abstract TSF representation provided is an accurate and complete instantiation of the TOE security functional requirements.

5.3.3.4 Descriptive low-level design (ADV_LLD.1)

ADV_LLD.1.1d The developer shall provide the low-level design of the TSF.
ADV_LLD.1.1c The presentation of the low-level design shall be informal.
ADV_LLD.1.2c The low-level design shall be internally consistent.
ADV_LLD.1.3c The low-level design shall describe the TSF in terms of modules.
ADV_LLD.1.4c The low-level design shall describe the purpose of each module.

ADV_LLD.1.5c The low-level design shall define the interrelationships between the modules in terms of provided security functionality and dependencies on other modules.

ADV_LLD.1.6c The low-level design shall describe how each TSP-enforcing function is provided.

ADV_LLD.1.7c The low-level design shall identify all interfaces to the modules of the TSF.

ADV_LLD.1.8c The low-level design shall identify which of the interfaces to the modules of the TSF are externally visible.

ADV_LLD.1.9c The low-level design shall describe the purpose and method of use of all interfaces to the modules of the TSF, providing details of effects, exceptions and error messages, as appropriate.

ADV_LLD.1.10c The low-level design shall describe the separation of the TOE into TSP-enforcing and other modules.

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

ADV_LLD.1.2e The evaluator shall determine that the low-level design is an accurate and complete instantiation of the TOE security functional requirements.

5.3.3.5 Informal correspondence demonstration (ADV_RCR.1)

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

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

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

5.3.3.6 Informal TOE security policy model (ADV_SPM.1)

ADV_SPM.1.1d The developer shall provide a TSP model.

ADV_SPM.1.2d The developer shall demonstrate correspondence between the functional specification and the TSP model.

ADV_SPM.1.1c The TSP model shall be informal.

ADV_SPM.1.2c The TSP model shall describe the rules and characteristics of all policies of the TSP that can be modeled.

ADV_SPM.1.3c The TSP model shall include a rationale that demonstrates that it is consistent and complete with respect to all policies of the TSP that can be modeled.

ADV_SPM.1.4c The demonstration of correspondence between the TSP model and the functional specification shall show that all of the security functions in the functional specification are consistent and complete with respect to the TSP model.

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

5.3.4 Guidance documents (AGD)

5.3.4.1 Administrator guidance (AGD_ADM.1)

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

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

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

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

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

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

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

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

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

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

5.3.4.2 User guidance (AGD_USR.1)

AGD_USR.1.1d The developer shall provide user guidance.

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

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

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

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

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

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

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

5.3.5 Life cycle support (ALC)

5.3.5.1 Identification of security measures (ALC_DVS.1)

ALC_DVS.1.1d The developer shall produce development security documentation.

ALC_DVS.1.1c The development security documentation shall describe 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.

ALC_DVS.1.2c The development security documentation shall provide evidence that these security measures are followed during the development and maintenance of the TOE.

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

ALC_DVS.1.2e The evaluator shall confirm that the security measures are being applied.

5.3.5.2 Flaw reporting procedures (ALC_FLR.2)

ALC_FLR.2.1d The developer shall provide flaw remediation procedures addressed to TOE developers.

ALC_FLR.2.2d The developer shall establish a procedure for accepting and acting upon all reports of security flaws and requests for corrections to those flaws.

ALC_FLR.2.3d The developer shall provide flaw remediation guidance addressed to TOE users.

ALC_FLR.2.1c The flaw remediation procedures documentation shall describe the procedures used to track all reported security flaws in each release of the TOE.

ALC_FLR.2.2c The flaw remediation procedures shall require that a description of the nature and effect of each security flaw be provided, as well as the status of finding a correction to that flaw.

ALC_FLR.2.3c The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.

ALC_FLR.2.4c The flaw remediation procedures documentation shall describe the methods used to provide flaw information, corrections and guidance on corrective actions to TOE users.

ALC_FLR.2.5c The flaw remediation procedures documentation shall describe a means by which the developer receives from TOE users reports and enquiries of suspected security flaws in the TOE.

ALC_FLR.2.6c The procedures for processing reported security flaws shall ensure that any reported flaws are corrected and the correction issued to TOE users.

ALC_FLR.2.7c The procedures for processing reported security flaws shall provide safeguards that any corrections to these security flaws do not introduce any new flaws.

ALC_FLR.2.8c The flaw remediation guidance shall describe a means by which TOE users report to the developer any suspected security flaws in the TOE.

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

5.3.5.3 Developer defined life-cycle model (ALC_LCD.1)

ALC_LCD.1.1d The developer shall establish a life-cycle model to be used in the development and maintenance of the TOE.

ALC_LCD.1.2d The developer shall provide life-cycle definition documentation.

ALC_LCD.1.1c The life-cycle definition documentation shall describe the model used to develop and maintain the TOE.

ALC_LCD.1.2c The life-cycle model shall provide for the necessary control over the development and maintenance of the TOE.

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

5.3.5.4 Well-defined development tools (ALC_TAT.1)

ALC_TAT.1.1d The developer shall identify the development tools being used for the TOE.

ALC_TAT.1.2d The developer shall document the selected implementation-dependent options of the development tools.

ALC_TAT.1.1c All development tools used for implementation shall be well-defined.

ALC_TAT.1.2c The documentation of the development tools shall unambiguously define the meaning of all statements used in the implementation.

ALC_TAT.1.3c The documentation of the development tools shall unambiguously define the meaning of all implementation-dependent options.

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

5.3.6 Tests (ATE)

5.3.6.1 Analysis of coverage (ATE_COV.2)

ATE_COV.2.1d The developer shall provide an analysis of the test coverage.

ATE_COV.2.1c The analysis of the test coverage shall demonstrate the correspondence between the tests identified in the test documentation and the TSF as described in the functional specification.

ATE_COV.2.2c The analysis of the test coverage shall demonstrate that the correspondence between the TSF as described in the functional specification and the tests identified in the test documentation is complete.

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

5.3.6.2 Testing: high-level design (ATE_DPT.1)

ATE_DPT.1.1d The developer shall provide the analysis of the depth of testing.

ATE_DPT.1.1c The depth analysis shall demonstrate that the tests identified in the test documentation are sufficient to demonstrate that the TSF operates in accordance with its high-level design.

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

5.3.6.3 Functional testing (ATE_FUN.1)

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

ATE_FUN.1.2d The developer shall provide test documentation.

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

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

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

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

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

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

5.3.6.4 Independent testing - sample (ATE_IND.2)

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

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

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

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

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

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

5.3.7 Vulnerability assessment (AVA)

5.3.7.1 Validation of analysis (AVA_MSU.2)

AVA_MSU.2.1d The developer shall provide guidance documentation.

AVA_MSU.2.2d The developer shall document an analysis of the guidance documentation.

AVA_MSU.2.1c The guidance documentation shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.

AVA_MSU.2.2c The guidance documentation shall be complete, clear, consistent and reasonable.

AVA_MSU.2.3c The guidance documentation shall list all assumptions about the intended environment.

AVA_MSU.2.4c The guidance documentation shall list all requirements for external security measures (including external procedural, physical and personnel controls).

AVA_MSU.2.5c The analysis documentation shall demonstrate that the guidance documentation is complete.

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

AVA_MSU.2.2e The evaluator shall repeat all configuration and installation procedures, and other procedures selectively, to confirm that the TOE can be configured and used securely using only the supplied guidance documentation.

AVA_MSU.2.3e The evaluator shall determine that the use of the guidance documentation allows all insecure states to be detected.

AVA_MSU.2.4e The evaluator shall confirm that the analysis documentation shows that guidance is provided for secure operation in all modes of operation of the TOE.

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

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

AVA_SOF.1.1c For each mechanism with a strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the minimum strength level defined in the PP/ST.

AVA_SOF.1.2c For each mechanism with a specific strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the specific strength of function metric defined in the PP/ST.

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

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

5.3.7.3 Independent vulnerability analysis (AVA_VLA.2)

AVA_VLA.2.1d The developer shall perform a vulnerability analysis.

AVA_VLA.2.2d The developer shall provide vulnerability analysis documentation.

AVA_VLA.2.1c The vulnerability analysis documentation shall describe the analysis of the TOE deliverables performed to search for ways in which a user can violate the TSP.

AVA_VLA.2.2c The vulnerability analysis documentation shall describe the disposition of identified vulnerabilities.

AVA_VLA.2.3c The vulnerability analysis documentation shall show, for all identified vulnerabilities, that the vulnerability cannot be exploited in the intended environment for the TOE.

AVA_VLA.2.4c The vulnerability analysis documentation shall justify that the TOE, with the identified vulnerabilities, is resistant to obvious penetration attacks.

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

AVA_VLA.2.2e The evaluator shall conduct penetration testing, building on the developer vulnerability analysis, to ensure the identified vulnerabilities have been addressed.

AVA_VLA.2.3e The evaluator shall perform an independent vulnerability analysis.

AVA_VLA.2.4e The evaluator shall perform independent penetration testing, based on the independent vulnerability analysis, to determine the exploitability of additional identified vulnerabilities in the intended environment.

AVA_VLA.2.5e The evaluator shall determine that the TOE is resistant to penetration attacks performed by an attacker possessing a low attack potential.

6. TOE Summary Specification

This chapter describes the security functions and associated assurance measures.

6.1 TOE Security Functions

6.1.1 Security Audit

The IDS includes an audit facility that can be enabled or disabled by an authorized administrator. When enabled, it generates records of security relevant events in accordance with audit masks configured by an authorized administrator. Global audit masks serve to define audit events that are always audited, never audited, and are audited by default for users that do not have specific masks. User specific masks define the audit events that will be audited for that specific user (in place of the global default mask). The audit configuration is managed using the *onaudit* utility program provided with the IDS.

The IDS can be configured to record audit events either in files The IDS includes a utility program, *onshowaudit*, that will extract the audit records into a file that can be reviewed using operating system tools (e.g., grep) or alternately can be loaded into an IDS database table using *dbload*. Once loaded, all of the IDS data manipulation commands (i.e., SQL statements) are at the disposal of the administrator to search and sort the audit records. Note that the resulting table readily allows searching and sorting based on user identity and event type as well as any other data found in the audit records (e.g., date and time). The process to load the audit records into IDS for review using IDS tools is described in the administrator guidance documents.

The IDS guidance includes procedures to ensure that audit files, as well as the rest of the TOE, are protected appropriately by the IT environment to ensure that only an authorized administrator can delete or otherwise access stored audit records.. It also includes instructions for configuring the TOE so that when audit records are loaded into an IDS table they will continue to be protected, using the IDS DAC mechanism, so that only the authorized administrator can access the audit information. Note that the IDS does not provide any ability within the TOE to modify audit records.

An authorized administrator can configure the IDS to stop auditing or stop the current SQL statement or other auditable event (effectively preventing auditable events) when the audit trail becomes full. Note that the TOE determines that the audit trail is full when an audit write fails to complete.

When the IDS records audit records, they contain the following information although some of the audit event types will contain more or less information as applicable:

- Timestamp – date and time of the audit event
- User name – identity of the responsible user
- Event Mnemonic– specific audit event name
- Database name – name of the applicable database (if any)
- Table identifier – identity of the applicable table (if any)
- Object name – name of the applicable object (if any)
- Success or failure – an indication of the success of the operation

  Note that the command line utilities generate audit records capturing the actual command line invocation. These records are generated directly by the utilities themselves.

The authorized administrator can configure The IDS to audit any or all of the available audit event types:

- Open, close, create, drop, grant, and revoke on a database;
- Alter, create, and drop on an index and a table;

- Create and drop on a view;
- Insert, update, select, and delete on a row;
- Create, drop, execute, and update statistics on a routine;
- Create and drop a synonym, type, or sequence;
- Use of privileged SQL statements;
- Change identity;
- Set logging mode;
- Update statistics; and
- Client connection request.

The Security Audit security function satisfies the following security requirements:

FAU_GEN.1a Audit data generation – The IDS fulfills this requirement by generating the necessary events associated with each of its security functions (and security functional requirements) and by including the date and time, event type, user identities, and results in each event.

FAU_GEN.2 User identity association – The IDS fulfills this requirement by including the applicable user identity in each audit record.

FAU_SAR.1 Audit review – The IDS fulfills this requirement by providing an interface for the review of audit records (*onshowaudit*).

FAU_SAR.2 Restricted audit review – The IDS fulfills this requirement by ensuring that the user is an authorized administrator (per their role) before allowing access to the audit records.

FAU_SAR.3 Selectable audit review – The IDS fulfills this requirement by providing search capabilities that can be realized by first exporting the audit records and then importing them back into a database table where arbitrary queries could be made (e.g., to search or sort based on user identity or event type).

FAU_SEL.1 Selective audit – The IDS fulfills this requirement by allowing an authorized administrator to configure IDS to audit any or all of the available audit event types both globally and for specific users.

FAU_STG.4 Prevention of audit data loss – The IDS fulfills this requirement by discarding auditable events that occur while the audit trail is full.

6.1.2 Access Control

The IDS discretionary access control (DAC) mechanism has the ability to include or exclude access to IDS objects on a per user basis and enables users to control other user’s access to these objects. No user can access the information in a database unless that user has been authorized explicitly or by default to access it in accordance with the DAC policy. Note that the IDS DAC mechanism is distinct from that of the underlying operating system, though it does rely on the underlying operating system to provide and protect its storage media (i.e., files).

The IDS DAC policy protects information stored in databases up to the granularity of individual columns within given tables. The IDS system catalog for a database includes tables that stored the access control list (ACL) that identifies users and their specific object access privileges within that database. The IDS system defines access privileges for databases, tables, views, synonyms, types, routines, and sequences. These access privileges can be granted and revoked using applicable SQL statements. They are granted to single users by username or alternately to all users using the special name ‘PUBLIC’.

Note that access privileges are organized by privilege levels associated with the applicable objects: database level privileges, table level privileges, type level privileges, routine level privileges, and sequence level privileges. Note that there are also language level privileges, but these are not considered security relevant as they serve only to limit the languages that can be used in user-defined routines. The privilege levels are really just logical groupings where the specific privileges in a given level can be assigned and serve to control access to the corresponding objects.

Tables, views, synonyms, types, routines, and sequences all have an owner which is the user that created the object. Also, if the object is created by a DBA, the object name can be specified as *user.objectname* and that user becomes the owner of the object. In order to access such an object, the user would still need at least CONNECT privilege to connect to the database where the object resides. A database, on the other hand, has a creator which automatically gets the 'dba' privilege, but there is no notion of a database owner. Administrative responsibility for a database can be changed by granting and revoking the dba privilege for that database. Basically, the dba privilege embodies the full range of privileges available for the database and any user possessing that privilege is referred to as the DBA for that database. Among other things, the DBA can designate (i.e., change) the owner of objects within their database.

A user cannot access any information in a database unless they have at least one of the following privileges for that database: dba, resource, or connect. The connect privilege enables a user to access the database, including the ability to store retrieved information in temporary tables. The resource privilege implies connect and further enables a user to create tables and indexes within the database. Finally, the dba privilege implies resource and connect and embodies full administrative authority in the database, including the ability to grant and revoke database privileges to another user. Note that each of these privileges applies only to a specific database. Also each privilege is checked when access is attempted (connect to a database, query a table, etc.). If a privilege should change, the change becomes effective the next time the user attempts an access that requires an access check.

A user can access information in a table only if that user has at least one of the following privileges for that table: alter, delete, index, insert, reference, select, and update. The alter privilege allows a user to change the relational schema of a table, as well as add or drop constraints on columns of the table. The alter privilege implies the index privilege which allows a user to create an index on a table. The delete privilege allows a user to delete a row from the table while the insert privilege allows a user to insert a new row into the table. The reference privilege allows a user to define referential constraints on the table. Note that the select, reference, and update privileges on a table can be granted on a certain column or columns in the associated table. The select privilege allows a user to retrieve data from all or some of the columns in the table. The update privilege implies the select privilege and allows a user to change values in some or all of the columns in the table.

DAC on views and synonyms is controlled by database and table privileges. There are no specific privileges defined for these objects.

In order to access a type the user must have usage privilege to use the type or under privilege in order to create a subtype for the type.

To execute a routine, a user is required to have execute privilege on the routine.

In order to use or alter a sequence, the user must have select or alter privilege, respectively.

The various object privileges have dependencies depending on the relationships among objects. The alter and index table privileges depend on the user having the associated resource database privilege. The delete, insert, select, and update table privileges depend on the user having the associated connect database privilege.

Note that there are actually three types of tables: permanent tables, as already discussed above, temporary tables, and views. Temporary tables are created to complete operations such as joins in order to return information to users. These tables are dropped when the user terminates their database session. In order to create a view, a user must have the connect database privilege on the applicable database as well as the select table privilege on the applicable table columns (or other views that may be used in creating the view). Note that views inherit access controls from the associated tables and views when created.

The IDS implements roles that can be specifically associated with users allowing them to exercise privileged SQL statements. As indicated later, any user that is assigned any of the administrative roles is considered in effect to be an authorized administrator since those roles each allow access to SQL statements that cannot be used by otherwise untrusted users. There are only a small number of roles, but the DBSA, for example, can bypass the normal DAC rules.

In addition to controlling access using permissions, authorized administrators can define LBAC security labels and authorized users can assign LBAC policies to tables. Once a LBAC policy is assigned to a table (the notion of 'protecting' a table), if the table contains a security label column the table is protected with row level granularity, otherwise if the table has a column protected with a security label (per the table definition) the table is protected with column level granularity. Subsequently, when a user attempts to create, modify, or otherwise access data in the table their access is restricted, in addition to the Discretionary Access Control rules, based on the security label associated with their session, the security label(s) associated with the table, and the LBAC access rules. Hence, the requested access to specific rows or columns is subject to the LBAC constraints.

LBAC labels have zero (0) or more of each of the three available component types (but must always have at least one component):

> **Array** – represents an ordered set; any element in the set is ranked higher than subsequent elements in the set.
>
> **Set** – represents an unordered set; there is no defined relationship among the elements in the set and there order is not important.
>
> **Tree** – represents a hierarchy and is used to represent organizational charts and to identify departments within an organization that owns the applicable data.

There are two sets of three rules that determine the allowed access based on LBAC labels:

> **Read Access Rules** apply when data is retrieved. Data is retrieved during SELECT, UPDATE, and DELETE operations.
>
> > **LBACREADARRAY** – Each array component of the user's security label must be greater than or equal to the corresponding array component of the data (row or column) security label.
> >
> > **LBACREADTREE** – Each tree component of the user's security label must include at least one of the elements in the corresponding tree component of the data (row or column) security label (or the ancestor of one such element).
> >
> > **LBACREADSET** – Each set component of the user's security label must include the corresponding set component of the data (row or column) security label.
>
> **Write Access Rules** apply for INSERT, UPDATE, and DELETE operations.
>
> > **LBACWRITEARRAY** – Each array component of the user's security label must be equal to the corresponding array component of the data (row or column) security label.
> >
> > **LBACWRITETREE** – Each tree component of the user's security label must include at least one of the corresponding elements in the tree component of the data (row or column) security label (or the ancestor of one such element).
> >
> > **LBACWRITESET** – Each set component of the user's security label must include the corresponding set component of the data (row or column) security label.

In addition to the rules cited above, IDS offers specific *exemptions* that can be assigned to users to bypass one or more of the read and write rules summarized above.

The IDS is designed to restrict access to objects until its resources have first been written. While the IDS does not actually clear resources, it ensures that information is not inappropriately reused or accessed by allowing data to be read after it has been written or initialized. There are many internal IDS resources that are carefully managed to prevent the possibility of inappropriate disclosure. As for externally accessible objects, the IDS manages free pages that are available and could be added to objects. When objects are created or extended, resources are added but cannot be read until they are used. When used, the applicable resource is initialized and its contents are managed to ensure that only previously written content can be read. When freed, resources are simply marked as free and are available for reuse.

The Access Control security function satisfies the following security requirements:

> *FDP_ACC.1 Subset access control* – The IDS fulfills this requirement by associating privileges with all operations applicable to each identified IDS object and requiring that a user have the privilege or an administrative role when attempting to perform the corresponding operation.

FDP_ACF.1 Security attribute based access control – The IDS fulfills this requirement by associating privileges with all operations applicable to each identified IDS object and requiring that a user have the privilege or an administrative role when attempting to perform the corresponding operation.

FDP_IFC.1 Subset information flow control – The IDS fulfills this requirement by allowing tables to be assigned LBAC policies that will control subsequent read and write operations.

FDP_IFF.2 Hierarchical security attributes – The IDS fulfills this requirements by enforcing the LBAC information flows rules as summarized above.

FDP_RIP.2a Full residual information protection – The IDS fulfills this requirement by ensuring that data can only be read after it has first been written.

6.1.3 Identification & Authentication

The IDS accepts connections both locally and across the network. In both cases, the IDS depends on the host operating system or a configured pluggable authentication module (PAM) to identify and authenticate the users and to provide the resulting username in order to appropriately associate the username with the resulting session. The username (and associated OS groups) also serves to allow the IDS to determine the user's roles, stored in the IDS, and also allows privileges to be looked up in ACLs in order to determine access privileges for specific objects. The username also allows IDS to determine the security label and any LBAC-related exemptions for the user's session.

When a session is initially created, the user's username and roles are associated with that session. Those attributes can be changed only by an authorized administrator using a privileged SQL statement.

Users can perform IDS functions in only one of two ways – they can establish a session with the server or they can exercise command line utilities in the context of the underlying operating system. In the latter case, the guidance serves to ensure that the applicable program and data files are appropriately configured and protected (by the IT environment) so they are appropriately controlled. Note, however, that each of the utilities is aware of the user's identity for the purpose of generating audit records and performing other functions.

The Identification & Authentication security function satisfies the following security requirements:

FIA_ATD.1a User attribute definition – The IDS fulfills this requirement by maintaining a correspondence between usernames (from the host operating system), roles, and LBAC security labels and exemptions.

FIA_UID.2a User identification before any action – The IDS fulfills this requirement by allowing access to IDS resources only when the user has been identified.

FIA_USB.1 User-subject binding – The IDS fulfills this requirement by associating usernames and roles with user sessions and allowing only an authorized user to change those attributes.

6.1.4 Security Management

The IDS identifies users in the authorized administrator role by the assignment of a specific administrative role (allowing access to privileged SQL statements) by the assignment of the user to specific groups in the IT environment. The IDS implements the functions associated with (i.e., restricted by) these roles and offers guidance for the configuration of the TOE in the host environment such that its program and data files are protected so that users must be in one of the specifically defined groups in order to act as an administrator of the TOE in the environment or within the TOE itself.

There are three defined roles for the TOE and an operating system role defined to operate in the TOE's environment (i.e., host operating system). The Operating System Administrator (OSA) is a logical notion of an administrator in the host operating system that has responsibilities for installing and managing aspects of the TOE's installation (e.g., to create the special groups and assign access permissions to various TOE program and data files). The Database System Security Officer (DBSSO) is responsible to manage the security properties (e.g., configuring audit) of IDS. The Database System Administrator (DBSA) is responsible to configure, tune, and monitor the IDS once it is operational. The Database Security Administrator (DBSECADM) is responsible to manage the LBAC policy, including the definition, modification, and assignment of security labels, policies, and exemptions. The Audit

Analysis Officer (AAO) is responsible to review and analyze the audit trail. The distinction of these roles is enforced by the IT environment and as such is not claimed in this ST; rather they are all treated simply as 'authorized administrator' though their access to privileged SQL statements may vary. Similarly, the single role defined in the IT environment is specific instance of the authorized administrator defined in the IT environment SFRs.

The TOE does, however, offer the utility programs (e.g., dbaccess) necessary to effectively perform in these roles. To that end there are utilities to enable and disable audit, configure audit selection masks, and review audit records (including importing them into IDS to use the query engine for better analysis capabilities). There are also utilities that allow security policies (and their components) to be created and security labels and exemptions to be granted and revoked to and from users and applicable database objects. Only authorized administrators can manage security labels and exemptions.

On the other hand, access to IDS objects is managed directly through the IDS itself. The IDS offers commands to manipulate object access control lists and ensures that access to objects is restricted when they are initial created. Note that these commands can be accessed by an interactive user using the dbaccss utility.

When a new object is created, the initial access is established by default. If the object is a database, the user who created it is given the dba privilege (i.e., is the DBA for that database). If the object is a table, the creating user becomes the owner and is given all table privileges for that table. If the object is a view, synonym, constraint, or index, the creating user becomes the owner but gets no specific privileges since none are defined. No user other than the creator initially has any privileges to the object with the exception of the DBA who has implicit privileges on tables created in the associated database.

Changes in the access privileges of a user to a given object occur when a user explicitly grants or revokes privileges to or from the user to the object. Grants and revocations of privileges do not necessarily take effect immediately. However, they do affect all future access decisions regarding the applicable object. In other words, an operation where an access check has already been made will complete regardless of any change in privilege but the next access check would reflect the change in privilege.

A user cannot grant or revoke a database privilege (dba, resource, connect) to or from another user unless that user has the dba privilege for that database.

A user, other than the table owner, will have a table privilege only if another user previously granted that privilege. Each table privilege can have a *grant* option that allows the user to give that privilege to another user. Table privileges can be granted with or without the grant option. Note that possession of the dba privilege implies that the user has all privileges on all tables in the associated database without the grant option. A user can only revoke a table privilege from another user if the revoking user originally granted the table privilege. The can result in a cascading revocation effect when a privilege is revoke from a user that granted the privilege to another user; when the first user's privilege is revoked it is also revoked for all users that user granted it to. Note that revocation is unaffected by whether the privilege was granted with or without the grant option. Note also that a user cannot revoke their own privileges.

When the select privilege is revoked on a table or view, any other views that are based on that table or view are automatically dropped. When any privilege other than select is lost on a table or view, that privilege is also revoked on any depending views. The revocation of privileges on a view does not affect the columns of the base table, however.

As for LBAC, users can assign labels to tables and their contents only in accordance with the LBAC rules as specified in section 6.1.2. IDS objects do not have labels by default; rather, they must be explicitly defined by users.

The Security Management security function satisfies the following security requirements:

> *FMT_MOF.1 Management of security functions behaviour* – The IDS fulfills this requirement by restricting the abilities to create and drop LBAC security labels, label components and policies as well as to grant and revoke security policies and LBAC exemptions to the authorized administrator.
>
> *FMT_MSA.1a Management of Security Attributes* – The IDS fulfills this requirement by allowing only users with the appropriate privileges to modify the Discretionary Access Control security attributes of any IDS object.

FMT_MSA.1a Management of Security Attributes – The IDS fulfills this requirement by allowing users to modify the LBAC security attributes of IDS objects only as allowed by the LBAC rules.

FMT_MSA.3a Static Attribute Initialization – The IDS fulfills this requirement by ensuring that objects are assigned restrictive default security attributes when created.

FMT_MSA.3a Static Attribute Initialization – The IDS fulfills this requirement by not assigning LBAC security attributes to objects by default, requiring those attributes to be explicitly assigned by authorized users.

FMT_REV.1a Revocation – The IDS fulfills this requirement by allowing only users with the appropriate privilege or administrative role to modify (including revoke) the security attributes of any IDS object.

FMT_SMF.1a Specification of Management Functions – The IDS fulfills this requirement by providing functions that allow an authorized administrator to start, stop, configure the audit security functions as well as the ability to review audit records, manage Discretionary Access Control privileges, and manage LBAC security policies and exemptions.

FMT_SMR.1a Security Management Roles – The IDS fulfills this requirement by defining authorized administrator and user roles, based on the association of administrative role with specific usernames by virtue of the user's OS group memberships.

6.1.5 TOE Protection

The IDS is designed to operate within a set of processes provided by the hosting operating system. The IDS does not support the ability to share its processes with non-TOE entities. Furthermore, the IDS is designed in a manner that ensures that its interfaces do not offer unauthorized users any functions that might be used to corrupt, or otherwise inappropriately access, the TSF. As is the case with many application-only TOEs such as the IDS, its protection mechanisms could be bypassed through the underlying environment should the assumptions (e.g., A.Platform) and requirements (e.g., FPT_SEP.1) for the IT environment not be fulfilled. Note that determination of fulfillment of those assumptions and IT environment requirements is not within the scope of the TOE.

The IDS has been designed to implement a number of IDS-specific objects and functions. Each IDS object and function is available via interfaces provided by the IDS, and each interface has been carefully designed to ensure that it only provides appropriate capabilities or access after necessary security checks have been made and approved.

The IDS has been designed to collect current time information from its hosting operating system in a correct and consistent manner. Once it has been collected, the IDS ensures that it is not corrupted as it is being used by the IDS TSF, thereby ensuring that it remains reliable.

The TOE Protection security function satisfies the following security requirements:

FPT_RVM.1a Non-bypassability of the TSP – The IDS fulfills this requirement by making sure that all applicable access checks are made by each of its interfaces before allowing access to IDS resources.

FPT_STM.1a Reliable Time Stamps – The IDS fulfills this requirement by consistently collecting time information from the IT environment and then by protecting it while it is being used. Note that a similar requirement is levied on the IT environment to ensure that it also has access reliable timestamps.

6.2 TOE Security Assurance Measures

6.2.1 Configuration management

The configuration management measures applied by IBM ensure that configuration items are uniquely identified, and that documented procedures are used to control and track changes that are made to the TOE. IBM ensures changes to the implementation representation are controlled with the support of automated tools and that TOE associated configuration item modifications are properly controlled. IBM performs configuration management on the TOE implementation representation, design documentation, tests and test documentation, user and administrator guidance, delivery and operation documentation, life-cycle documentation, vulnerability analysis documentation, configuration management documentation, and security flaws.

These activities are documented in:

- IBM IDS Server Version 11.5 Configuration Management Plan

The Configuration management assurance measure satisfies the following EAL 4 augmented with ALC_FLR.2 assurance requirements:

- ACM_AUT.1
- ACM_CAP.4
- ACM_SCP.2

6.2.2 Delivery and operation

IBM provides delivery documentation and procedures to identify the TOE, allow detection of unauthorized modifications of the TOE and installation and generation instructions at start-up. IBM's delivery procedures describe all applicable procedures to be used to detect modification to the TOE. IBM also provides documentation that describes the steps necessary to install IDS in accordance with the evaluated configuration.

These activities are documented in:

- IBM IDS 11.50 Delivery Procedures
- IBM IDS 11.50 Common Criteria Certification: Requirements for Informix Dynamic Server

The Delivery and operation assurance measure satisfies the following EAL 4 augmented with ALC_FLR.2 assurance requirements:

- ADO_DEL.2
- ADO_IGS.1

6.2.3 Development

IBM has numerous documents describing all facets of the design of the TOE. In particular, they have a functional specification that describes the accessible TOE interfaces; a high-level design that decomposes the TOE architecture into subsystems and describes each subsystem and their interfaces; a low-level design that further decomposes the TOE architecture into modules and describes each module and their interfaces; and, correspondence documentation that explains how each of the design abstractions correspond from the TOE summary specification in the Security Target to the actual implementation of the TOE. Furthermore, IBM has a security model that describes each of the security policies implemented by IDS. Of course, the implementation of the TOE itself is also available as necessary.

These activities are documented in:

- IBM Corporation Informix Dynamic Server Version 11.5 Functional Specification
- IBM Corporation Informix Dynamic Server Version 11.5 High Level Design
- IBM Corporation Informix Dynamic Server Version 11.5 Low Level Design
- IBM Corporation IDS 11.5 Security Policy Model
- IBM Informix Dynamic Server source code

The Development assurance measure satisfies the following EAL 4 augmented with ALC_FLR.2 assurance requirements:

- ADV_FSP.2
- ADV_HLD.2

- ADV_IMP.1
- ADV_LLD.1
- ADV_RCR.1
- ADV_SPM.1

6.2.4 Guidance documents

IBM provides administrator and user guidance on how to utilize the TOE security functions and warnings to administrators and users about actions that can compromise the security of the TOE.

These activities are documented in:

- IBM IDS 11.50 Common Criteria Certification: Requirements for Informix Dynamic Server
- IBM Informix, Version 11.50, IBM Informix Dynamic Server Administrator's Guide
- IBM Informix, Version 11.50, IBM Informix Security Guide
- IBM Informix, Version 11.50, IBM Informix Guide to SQL: Syntax

The Guidance documents assurance measure satisfies the following EAL 4 augmented with ALC_FLR.2 assurance requirements:

- AGD_ADM.1
- AGD_USR.1

6.2.5 Life cycle support

IBM ensures the adequacy of the procedures used during the development and maintenance of the TOE through the use of a comprehensive life-cycle management plan. IBM applies security controls on the development environment that are adequate to provide the confidentiality and integrity of the TOE design and implementation that is necessary to ensure the secure development of the TOE. IBM has procedures that define the process for accepting and acting upon user reports of security flaws. These procedures describe the acceptance criteria for security flaws, how all security flaws and the status of fixes for each security flaw are tracked, and how corrections and corrective measures are made available as applicable. IBM has a documented model of the TOE life cycle that ensures that the TOE is developed and maintained in a well-defined manner. IBMuses well-defined development tools in order to ensure consistent and predictable results while developing the TOE.

These activities are documented in:

- IBM IDS SERVER 11.10 5 for Linux, UNIX, and Windows Life Cycle Document

The Life cycle support assurance measure satisfies the following EAL 4 augmented with ALC_FLR.2 assurance requirements:

- ALC_DVS.1
- ALC_FLR.2
- ALC_LCD.1
- ALC_TAT.1

6.2.6 Tests

IBM has a test plan that describes how each of the necessary security functions is tested, along with the expected test results. IBM has documented each test as well as an analysis of test coverage and depth demonstrating that the security aspects of the design evident from the functional specification and high-level design are appropriately tested. Actual test results are created on a regular basis to demonstrate that the tests have been applied and that the TOE operates as designed.

These activities are documented in:

- IBM IDS Version 11.5 For Linux, Unix, and Windows Test Plan
- IBM Corporation Informix Dynamic Server Version 11.5 Test Description
- IBM Corporation Informix Dynamic Server Version 11.5 Test Instruction
- IBM Corporation Informix Dynamic Server Version 11.5 Identification and Authentication Test
- Test code and results

The Tests assurance measure satisfies the following EAL 4 augmented with ALC_FLR.2 assurance requirements:

- ATE_COV.2
- ATE_DPT.1
- ATE_FUN.1
- ATE_IND.2

6.2.7 Vulnerability assessment

The TOE administrator and user guidance documents describe the operation of IDS and how to maintain a secure state. These guides also describe all necessary operating assumptions and security requirements outside the scope of control of the TOE. They have been developed to serve as complete, clear, consistent, and reasonable administrator and user references. Furthermore, IBM has conducted a misuse analysis demonstrating that the provided guidance is complete.

IBM has conducted a strength of function analysis where it has been determined that the TOE contains no permutational or probabilistic security mechanisms and as a result the minimum strength of function claim, SOF-medium is not particularly applicable.

IBM performs regular vulnerability analyses of the entire TOE (including documentation) to identify weaknesses that can be exploited in the TOE.

These activities are documented in:

- IBM Informix Dynamic Server (IDS) Enterprise Edition Vulnerability Assessment
- IBM Informix Dynamic Server (IDS) Enterprise Edition Misuse Analysis

The Vulnerability assessment assurance measure satisfies the following EAL 4 augmented with ALC_FLR.2 assurance requirements:

- AVA_MSU.2
- AVA_SOF.1
- AVA_VLA.2

7. Protection Profile Claims

There are no Protection Profile claims in this Security Target.

8. Rationale

This section provides the rationale for completeness and consistency of the Security Target. The rationale addresses the following areas:

- Security Objectives;
- Security Functional Requirements;
- Security Assurance Requirements;
- Strength of Functions;
- Requirement Dependencies;
- TOE Summary Specification; and,
- PP Claims.

8.1 Security Objectives Rationale

This section provides a rationale for the existence of each threat, policy statement, security objective, and component that comprise the protection profile.

8.1.1 Complete Coverage - Threats

The TOE security objectives have been derived exclusively from statements of organizational security policy, and therefore, there are no explicitly defined threats countered by this profile.

8.1.2 Complete Coverage - Policy

This section provides evidence demonstrating coverage of the Organizational Security Policy by both the IT and Non-IT security objectives. The following table shows this objective to policy mapping, and the table is followed by a discussion of the coverage for each Security Policy.

Organizational Security Policy	Security Objectives
P.AUTHORIZED_USERS	O.AUTHORIZATION
	OE.AUTHORIZATION
	O.MANAGE
	OE.MANAGE
	O.ENFORCEMENT
	OE.ENFORCEMENT
P.NEED_TO_KNOW	O.DISCRETIONARY_ACCESS
	O.RESIDUAL_INFORMATION
	OE.RESIDUAL_INFORMATION
	O.MANAGE
	OE.MANAGE
	O.ENFORCEMENT
	OE.ENFORCEMENT
P.ACCOUNTABILITY	O.AUDITING
	OE.AUDITING
	O.MANAGE
	OE.MANAGE
	O.ENFORCEMENT
	OE.ENFORCEMENT

Organizational Security Policy	Security Objectives
P.CLASSIFICATION	O.MANDATORY_ACCESS
	O.RESIDUAL_INFORMATION
	O.MANAGE
	O.ENFORCEMENT

Table 5 Mapping of Organizational Security Policies to Security Objectives

The following discussion provides detailed evidence of coverage for each statement of organizational security policy:

P.AUTHORIZED_USERS

Only those users who have been authorized to access the information within the TOE may access the TOE.

This policy is primarily realized by the O.AUTHORIZATION and OE.AUTHORIZATION objectives. The O.AUTHORIZATION and OE.AUTHORIZATION objectives require that the TOE and IT environment provide access only to authorized users. The O.MANAGE and OE.MANAGE objectives support this policy by requiring that an authorized administrator is able to manage the functions. The O.ENFORCEMENT and OE.ENFORCEMENT objectives ensure that functions are invoked and operate correctly.

P.NEED_TO_KNOW

The TOE must limit the access to, modification of, and destruction of the information in protected resources to those authorized users which have a "need to know" for that information.

This policy is primarily realized by the O.DISCRETIONARY_ACCESS objective, which allows authorized users to control access to resources based on user identities. The O.RESIDUAL_INFORMATION and OE.RESIDUAL_INFORMATION objectives ensure that information will not be given to users that do not have a need-to-know when resources are reused. The O.MANAGE and OE.MANAGE objectives support this policy by requiring that an authorized administrator is able to manage the functions. The O.ENFORCEMENT and OE.ENFORCEMENT objectives ensure that functions are invoked and operate correctly.

P.ACCOUNTABILITY

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

This policy is primarily realized by the O.AUDITING and OE.AUDITING objectives by requiring that actions are recorded in an audit trail. The O.MANAGE and OE.MANAGE objectives support this policy by requiring that an authorized administrator is able to manage the functions. The O.ENFORCEMENT and OE.ENFORCEMENT objectives ensure that functions are invoked and operate correctly.

P.CLASSIFICATION

The system must be able to limit the access to information based on sensitivity, as represented by a label, of the information contained in objects, and the formal clearance of users, as represented by subjects, to access that information. The access rules enforced prevent a subject from accessing information which is of higher sensitivity than it is operating at.

This policy is implemented by the O.MANDATORY_ACCESS objective. The O.RESIDUAL_INFORMATION objective ensures that information will not given to users which do not have a cleared access, when resources are reused. The O.MANAGE supports this policy by requiring authorized administrator be able to manage the functions and O.ENFORCEMENT ensures that functions are invoked and operate correctly.

8.1.3 Complete Coverage - Environmental Assumptions

This section provides evidence demonstrating coverage of the Non-IT security objectives by the environmental assumptions. The following table shows this assumption to objective mapping.

Environmental Assumptions	Non-IT Security Objectives
A.MANAGE	O.ASSIGN
	O.INSTALL
A.NO_EVIL_ADM	O.ADMIN_GUIDANCE
	O.ADMINISTRATORS
	O.INSTALL
A.LOCATE	O.PHYSICAL
A.PROTECT	
A.CONNECT	
A.COOP	O.COOP
	O.CREDEN
A.PLATFORM	O.PLATFORM
A.CLEARANCE	O.CREDEN

Table 6 Mapping of Environmental Assumptions to Non-IT Security Objectives

A.MANAGE

There will be one or more competent individuals assigned to manage the TOE and the security of the information it contains.

This is addressed by O.ASSIGN, which ensures that competent individuals are assigned to manage the TOE and the security of its information, and by O.INSTALL, which ensures that the TOE is delivered, installed, managed and operated in a manner that maintains IT security.

A.NO_EVIL_ADM

The system administrative personnel are not careless, willfully negligent, or hostile, and will follow and abide by the instructions provided by the administrator documentation.

This is primarily addressed by O.ADMINISTRATORS, which ensures that Administrators of the TOE and IT Environment must not be careless, willfully negligent or hostile, and must follow the instructions provided in the administrator guidance documentation. The O.ADMIN_GUIDANCE objective ensures that administrators receive guidance documentation enabling them to install, manage, and operate the TOE securely. This assumption is also addressed by O.INSTALL, which ensures that the TOE is delivered, installed, managed and operated in a manner that maintains IT security.

A.LOCATE

The processing resources of the TOE will be located within controlled access facilities, which will prevent unauthorized physical access.

This is addressed by O.PHYSICAL which addresses those parts of the TOE which are critical to security policy are protected from physical attack.

A.PROTECT

The hardware and software critical to security policy enforcement will be protected from unauthorized physical modification.

This is addressed by O.PHYSICAL which addresses those parts of the TOE which are critical to security policy are protected from physical attack.

A.CONNECT

All connections to peripheral devices reside within the controlled access facilities. The TOE only addresses security concerns related to the manipulation of the TOE through its authorized access points. Internal communication paths to access points such as terminals are assumed to be adequately protected.

This is addressed by O.PHYSICAL which ensures that those parts of the TOE critical to security policy are protected from physical attack that might compromise IT security objectives.

A.COOP

Authorized users possess the necessary authorization to access at least some of the information managed by the TOE and are expected to act in a cooperating manner in a benign environment.

This is addressed by O.COOP, which ensures that authorized users possess the appropriate authorization to access at least some of the information managed by the TOE and act in a cooperative manner in a benign environment. This is also addressed by O.CREDEN that states that those responsible for the TOE must ensure that all access credentials such as passwords or other authentication information are protected by the users in a manner that maintains IT security objectives.

A.PLATFORM

The IT Environment underlying the TOE is assumed to fulfill the requirements for the IT Environment described in this Security Target. It is also assumed that the IT Environment will provide a suitable operational environment for the TOE where the TOE will be able to properly execute and the dependencies that the TOE has upon the IT Environment are properly fulfilled.

This is addressed by O.PLATFORM that basically reiterates the assumption to expect the IT Environment to provide a suitable and effective environment for the operation of the TOE.

A.CLEARANCE

Procedures exist for granting users authorization for access to specific security levels. It is further assumed the TOE administrators will be cleared to the highest security level processed by the TOE.

This is addressed by O.CREDEN that states that credentials such as clearances, perhaps represented by security labels, must be associated with user appropriately.

8.2 Security Requirements Rationale

This section provides evidence supporting the combined internal consistency and completeness of the requirements in this Security Target.

8.2.1 Internal Consistency of Requirements

This section describes the mutual support and internal consistency of the components selected for this Security Target. These properties are discussed for both functional and assurance components.

The functional components were selected from pre-defined CC components. Assignment, selection, and refinement operations were carried out among components using consistent computer security terminology. This helps to avoid the ambiguity associated with interpretations of meanings of terms between related components. Multiple instantiation of components was used to clearly state the required functionality that must exist in the TOE.

Each security functional requirement in the ST was selected to avoid conflicts with other security functional requirements in the ST.

The IT security functional requirements form a mutually supportive whole. Table 8 in Section 8.2.2 maps the functional components to security objectives. Table 9 in Section 8.4 demonstrates that the TOE security functional requirement dependencies have been satisfied.

Additionally, Section 5 of the ST contains several security functional requirements that support other requirements, as detailed in the following table.

Security functional requirement	Effect
FAU_GEN.1	Detect attempts to bypass or tamper with other security functional requirements
FAU_GEN.2	
FAU_SAR.1	
FAU_STG.4	
FPT_RVM.1	Prevent other security functional requirements from being bypassed
FPT_SEP.1 FAU_STG.1	Prevent other security functional requirements from being tampered with
FPT_STM.1a FPT_STM.1b	Provide time stamps from the IT environment for required use by the TOE

Table 7 Mapping of Requirements to Effects

8.2.2 Complete Coverage - Objectives

This section demonstrates that the functional components selected for this Security Target provide complete coverage of the defined IT security objectives. The mapping of components to IT security objectives is depicted in the following table.

Security Objective	Functional Component
O.AUTHORIZATION	FIA_ATD.1a
	FIA_UID.2a
OE.AUTHORIZATION	FIA_ATD.1b
	FIA_SOS.1
	FIA_UAU.2
	FIA_UAU.7
	FIA_UID.2b
	FMT_MTD.1d
	FMT_MTD.1e
O.DISCRETIONARY_ACCESS	FDP_ACC.1
	FDP_ACF.1
	FIA_ATD.1a
	FIA_USB.1
	FMT_MSA.1a
	FMT_MSA.3a
	FMT_REV.1a
O.MANDATORY_ACCESS	FDP_IFC.1
	FDP_IFF.2
	FIA_ATD.1a

Security Objective	Functional Component
	FIA_USB.1
	FMT_MOF.1
	FMT_MSA.1b
	FMT_MSA.3b
O.AUDITING	FAU_GEN.1a
	FAU_GEN.2
	FAU_SAR.1
	FAU_SAR.2
	FAU_SAR.3
	FAU_SEL.1
	FAU_STG.4
	FIA_USB.1
	FMT_SMF.1a
	FPT_STM.1a
OE.AUDITING	FAU_GEN.1b
	FPT_STM.1b
O.RESIDUAL_INFORMATION	FDP_RIP.2a
OE.RESIDUAL_INFORMATION	FDP_RIP.2b
O.MANAGE	FAU_SAR.1
	FAU_SAR.3
	FAU_SEL.1
	FAU_STG.4
	FMT_SMF.1a
	FMT_SMR.1a
OE.MANAGE	FMT_MTD.1a
	FMT_MTD.1b
	FMT_MTD.1c
	FMT_MTD.1d
	FMT_MTD.1e
	FMT_REV.1b
	FMT_SMF.1b
	FMT_SMR.1b
O.ENFORCEMENT	FPT_RVM.1a
OE.ENFORCEMENT	FAU_STG.1
	FPT_AMT.1
	FPT_RVM.1b
	FPT_SEP.1
	FPT_STM.1b

Table 8 Mapping of Security Objectives to Functional Components

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

O.AUTHORIZATION

The TSF must ensure that only authorized users gain access to the TOE and its resources.

Users must be identified [FIA_UID.2a] and associated with available authorities and privileges [FIA_ATD.1a] before they can access the TOE and the resources it protects.

OE.AUTHORIZATION

The IT Environment must ensure that only authorized users gain access to the IT Environment and its resources. The IT Environment must support the TOE by ensuring that users are adequately authenticated on the TOE's behalf.

Users must be identified [FIA_UID.2b], authenticated [FIA_UAU.2], and associated with available roles and privileges [FIA_ATD.1b] before they can access the IT Environment and the resources it protects. Furthermore, the authentication data must be protected [FIA_UAU.7, FMT_MTD.1d, FMT_MTD.1e] and the authentication mechanism must have suitable strength [FIA_SOS.1].

O.DISCRETIONARY_ACCESS

The TSF must control access to resources based on identity of users. The TSF must allow authorized users to specify which users may access which resources.

Discretionary access control must have a defined scope of control [FDP_ACC.1]. The rules of the DAC policy must be defined [FDP_ACF.1]. The security attributes of objects used to enforce the DAC policy must be defined [FDP_ACF.1]. Authorized users must be able to control who has access to objects [FMT_MSA.1a] and be able to revoke that access [FMT_REV.1a]. Default protection must be available from an object's creation [FMT_MSA.3a].

O.MANDATORY_ACCESS

The TSF must be able to control access to resources based upon the sensitivity and categories of the information being accessed and the clearance of the subject attempting to access that information.

Mandatory access control attributes and rules must be definable [FDP_IFF.2] and must have a definable scope of control [FDP_IFC.1]. Finally, if the MAC policy is to be enforced, it is required that it can be enabled and that attributes be associated with each object [FMT_MOF.1, FMT_MSA.1b, FMT_MSA.3b], and that the binding between processes and the attributes of the user on whose behalf they operate be correct and unforgable [FIA_ATD.1a, FIA_USB.1].

O.AUDITING

The TSF must record the security relevant actions of users of the TOE. The TSF must present this information to authorized administrators.

Security-relevant actions must be defined, auditable [FAU_GEN.1a], and capable of being associated with individual users [FAU_GEN.2, FIA_USB.1]. The audit trail must be protected so that only authorized users may access it [FAU_SAR.2]. The TSF must provide the capability to audit specific types of actions [FAU_SEL.1] and the actions of individual users [FAU_SAR.3, FIA_USB.1]. The audit facility must have some defined behavior if the audit trail becomes full [FAU_STG.4]. The time stamp associated must be reliable [FPT_STM.1a]. An authorized administrator must be able to review [FAU_SAR.1] and manage [FMT_SMF.1a] the audit trail.

OE.AUDITING

The IT Environment must record the security relevant actions of users of the IT Environment.

Security-relevant actions in the IT environment must be defined and auditable in the IT environment [FAU_GEN.1b] and the audit records must have reliable time stamps [FPT_STM.1b].

O.RESIDUAL_INFORMATION

The TSF must ensure that any information contained in a protected resource is not released when the resource is recycled.

Residual information associated with defined objects in the TOE must be inaccessible during reuse of the object containing the residual information [FDP_RIP.2a].

OE.RESIDUAL_INFORMATION

The IT Environment must ensure that any information contained in a protected resource is not released when the resource is recycled.

Residual information associated with defined objects in the TOE, as realized using objects provided to the TOE from the IT environment (e.g., files), must be purged prior to the reuse of the object containing the residual information [FDP_RIP.2b].

O.MANAGE

The TSF must provide all the functions and facilities necessary to support the authorized administrators that are responsible for the management of TOE security.

The TSF must provide for an authorized administrator to manage the TOE [FMT_SMR.1a]. The administrator must be able to review and manage the audit trail [FAU_SAR.1, FAU_SAR.3, FAU_SEL.1, FAU_STG.4, FMT_SMF.1a] along with all other security functions of the TOE [FMT_SMF.1a].

OE.MANAGE

The IT Environment must provide all the functions and facilities necessary to support the authorized administrators that are responsible for the management of IT Environment security, including security relevant support for the TOE.

The IT Environment must provide for an authorized administrator to manage the IT Environment [FMT_SMR.1b]. The administrator must be able to administer user accounts [FMT_MTD.1c, FMT_MTD.1d, FMT_MTD.1e, FMT_REV.1b, FMT_SMF.1b]. The administrator must be able to manage the audit function of the TOE [FMT_MTD.1a and FMT_MTD.1b].

O.ENFORCEMENT

The TSF must be designed and implemented in a manner that ensures that the organizational policies are enforced in the target environment.

The TSF must make and enforce the decisions of its security policies [FPT_RVM.1a]. The correctness of this objective is further met through the assurance requirements defined in this Security Target. This objective provides global support to other security objectives for the TOE by protecting the parts of the TOE, which implement policies and ensures that policies are enforced.

OE.ENFORCEMENT

The IT Environment must be designed and implemented in a manner that ensures that it can protect the operational IT Environment of the TOE. The IT Environment must provide a reliable time source for the use of both the TOE and the IT Environment.

The IT Environment must make and enforce the decisions of its security policies [FPT_RVM.1b]. It must be protected from interference that would prevent it from performing its security functions [FPT_SEP.1]. Additionally, the IT Environment must provide the capability to demonstrate correct operation of the underlying abstract machine [FPT_AMT.1]. The IT Environment must also supply reliable time stamps [FPT_STM.1b] and protect stored audit data [FAU_STG.1].

8.3 Security Assurance Requirements Rationale

The TOE was developed based on the C2 requirements of the Trusted Computer System Evaluation Criteria (TCSEC). Those requirements have been reproduced in the Controlled Access Protection Profile (CAPP) using

Common Criteria conventions. While the CAPP demands only EAL 3, this Security Target claims EAL 4 augmented with ALC_FLR.2. This added assurance is intended to provide consumers more confidence in the security features of the TOE so that the product may be used in a wider variety of environments.

8.4 Strength of Functions Rationale

Although an explicit requirement for the strength of secrets (FIA_SOS.1) is assigned to the IT environment, there are no TOE security functional requirements or security mechanisms that are permutational or probabilistic in nature. Therefore, of the minimum SOF claim of SOF-medium does not apply to any claim about the TOE made in this Security Target, though it could be applied to the IT environment of the TOE.

8.5 Requirement Dependency Rationale

The following table shows the security functional and assurance requirement dependencies that exist based on the security functional and assurance requirements (and iterations thereof) included in this Security Target. As indicated in the following table all of the dependencies are satisfied.

Note that in the left column TOE security functional requirements are identified normally, IT environment security functional requirements are *italicized*, and assurance requirements are underlined.

ST Requirement	CC Dependencies	ST Dependencies
FAU_GEN.1a	FPT_STM.1	FPT_STM.1a/*FPT_STM.1b*
FAU_GEN.2	FAU_GEN.1 and FIA_UID.1	FAU_GEN.1a and FIA_UID.2a
FAU_SAR.1	FAU_GEN.1	FAU_GEN.1a
FAU_SAR.2	FAU_SAR.1	FAU_SAR.1
FAU_SAR.3	FAU_SAR.1	FAU_SAR.1
FAU_SEL.1	FAU_GEN.1 and FMT_MTD.1	FAU_GEN.1a and FMT_MTD.1b
FAU_STG.4	FAU_STG.1	*FAU_STG.1*
FDP_ACC.1	FDP_ACF.1	FDP_ACF.1
FDP_ACF.1	FDP_ACC.1 and FMT_MSA.3	FDP_ACC.1 and FMT_MSA.3a
FDP_IFC.1	FDP_IFF.1	FDP_IFF.2
FDP_IFF.2	FDP_IFC.1 and FMT_MSA.3	FDP_IFC.1 and FMT_MSA.3b
FDP_RIP.2a	none	none
FIA_ATD.1a	none	none
FIA_UID.2a	none	none
FIA_USB.1	FIA_ATD.1	FIA_ATD.1a
FMT_MOF.1	FMT_SMR.1 and FMT_SMF.1	FMT_SMR.1a and FMT_SMF.1a
FMT_MSA.1a	FMT_SMR.1 and FMT_SMF.1 and (FDP_ACC.1 or FDP_IFC.1)	FMT_SMR.1a and FMT_SMF.1a and FDP_ACC.1
FMT_MSA.1b	FMT_SMR.1 and FMT_SMF.1 and (FDP_ACC.1 or FDP_IFC.1)	FMT_SMR.1a and FMT_SMF.1a and FDP_IFC.1
FMT_MSA.3a	FMT_MSA.1 and FMT_SMR.1	FMT_MSA.1a and FMT_SMR.1a
FMT_MSA.3b	FMT_MSA.1 and FMT_SMR.1	FMT_MSA.1b and FMT_SMR.1a
FMT_REV.1a	FMT_SMR.1	FMT_SMR.1a
FMT_SMF.1a	none	none
FMT_SMR.1a	FIA_UID.1	FIA_UID.2a
FPT_RVM.1a	none	none
FPT_STM.1a	none	none
FAU_GEN.1b	FPT_STM.1	*FPT_STM.1b*
FAU_STG.1	FAU_GEN.1	FAU_GEN.1a/*FAU_GEN.1b*
FDP_RIP.2b	none	none
FIA_ATD.1b	none	none

ST Requirement	CC Dependencies	ST Dependencies
FIA_SOS.1	none	none
FIA_UAU.2	FIA_UID.1	*FIA_UID.2b*
FIA_UAU.7	FIA_UAU.1	*FIA_UAU.2*
FIA_UID.2b	none	none
FMT_MTD.1a	FMT_SMR.1 and FMT_SMF.1	*FMT_SMR.1b* and FMT_SMF.1a
FMT_MTD.1b	FMT_SMR.1 and FMT_SMF.1	*FMT_SMR.1b* and FMT_SMF.1a
FMT_MTD.1c	FMT_SMR.1 and FMT_SMF.1	*FMT_SMR.1b* and *FMT_SMF.1b*
FMT_MTD.1d	FMT_SMR.1 and FMT_SMF.1	*FMT_SMR.1b* and *FMT_SMF.1b*
FMT_MTD.1e	FMT_SMR.1 and FMT_SMF.1	*FMT_SMR.1b* and *FMT_SMF.1b*
FMT_REV.1b	FMT_SMR.1	*FMT_SMR.1b*
FMT_SMF.1b	none	none
FMT_SMR.1b	FIA_UID.1	*FIA_UID.2b*
FPT_AMT.1	none	none
FPT_RVM.1b	none	none
FPT_SEP.1	none	none
FPT_STM.1b	none	none
ACM_AUT.1	ACM_CAP.3	ACM_CAP.4
ACM_CAP.4	ALC_DVS.1	ALC_DVS.1
ACM_SCP.2	ACM_CAP.3	ACM_CAP.4
ADO_DEL.2	ACM_CAP.3	ACM_CAP.4
ADO_IGS.1	AGD_ADM.1	AGD_ADM.1
ADV_FSP.2	ADV_RCR.1	ADV_RCR.1
ADV_HLD.2	ADV_FSP.1 and ADV_RCR.1	ADV_FSP.2 and ADV_RCR.1
ADV_IMP.1	ADV_LLD.1 and ADV_RCR.1 and ALC_TAT.1	ADV_LLD.1 and ADV_RCR.1 and ALC_TAT.1
ADV_LLD.1	ADV_HLD.2 and ADV_RCR.1	ADV_HLD.2 and ADV_RCR.1
ADV_RCR.1	none	none
ADV_SPM.1	ADV_FSP.1	ADV_FSP.2
AGD_ADM.1	ADV_FSP.1	ADV_FSP.2
AGD_USR.1	ADV_FSP.1	ADV_FSP.2
ALC_DVS.1	none	none
ALC_FLR.1	none	none
ALC_LCD.1	none	none
ALC_TAT.1	ADV_IMP.1	ADV_IMP.1
ATE_COV.2	ADV_FSP.1 and ATE_FUN.1	ADV_FSP.2 and ATE_FUN.1
ATE_DPT.1	ADV_HLD.1 and ATE_FUN.1	ADV_HLD.2 and ATE_FUN.1
ATE_FUN.1	none	none
ATE_IND.2	ADV_FSP.1 and AGD_ADM.1 and AGD_USR.1 and ATE_FUN.1	ADV_FSP.2 and AGD_ADM.1 and AGD_USR.1 and ATE_FUN.1
AVA_MSU.2	ADO_IGS.1 and ADV_FSP.1 and AGD_ADM.1 and AGD_USR.1	ADO_IGS.1 and ADV_FSP.2 and AGD_ADM.1 and AGD_USR.1
AVA_SOF.1	ADV_FSP.1 and ADV_HLD.1	ADV_FSP.2 and ADV_HLD.2
AVA_VLA.2	ADV_FSP.1 and ADV_HLD.2 and ADV_IMP.1 and ADV_LLD.1 and AGD_ADM.1 and AGD_USR.1	ADV_FSP.2 and ADV_HLD.2 and ADV_IMP.1 and ADV_LLD.1 and AGD_ADM.1 and AGD_USR.1

Table 9 Security Requirement Dependencies

While the FAU_GEN.2, FMT_SMR.1a, FIA_UAU.2, and FMT_SMR.1b requirements are dependent upon the FIA_UID.1 requirement, the FIA_UID.2 requirement is used in this ST. Note that FIA_UID.2 is hierarchical to FIA_UID.1.

8.6 Explicitly Stated Requirements Rationale

This Security Target contains one explicitly stated requirement: FPT_STM.1a. This requirement is based on the CC version of FPT_STM.1, except that the explicitly stated version specifically allows the IT environment to perform some aspect of the requirement which is not allowed in the original requirement. In this case, the IT environment provides timestamps that are subsequently collected, protected, and used by the TOE. Note that the function implied by this requirement is completely fulfilled by a combination of the TOE and its IT environment and as such should be considered to satisfy any dependencies levied on FPT_STM.1.

8.7 TOE Summary Specification Rationale

Each subsection in Section 6, the TOE Summary Specification, describes a security function of the TOE. Each description is followed with rationale that indicates which 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 Section 6, the TOE Summary Specification, provides evidence that the security functions are suitable to meet the TOE security requirements. The collection of security functions work together to provide all of the security requirements. The security functions described in the TOE summary specification are all necessary for the required security functionality in the TSF. **Table 10 Security Functions vs. Requirements Mapping** demonstrates the relationship between security requirements and security functions.

Security Function	Security Functional Components
Security Audit	FAU_GEN.1a Audit data generation
	FAU_GEN.2 User identity association
	FAU_SAR.1 Audit review
	FAU_SAR.2 Restricted audit review
	FAU_SAR.3 Selectable audit review
	FAU_SEL.1 Selective audit
	FAU_STG.4 Prevention of audit data loss
Access Control	FDP_ACC.1 Subset access control
	FDP_ACF.1 Security attribute based access control
	FDP_IFC.1 Subset information flow control
	FDP_IFF.2 Hierarchical security attributes
	FDP_RIP.2a Full residual information protection
Identification & authentication	FIA_ATD.1a User attribute definition
	FIA_UID.2a User identification before any action
	FIA_USB.1 User-subject binding
Security management	FMT_MOF.1 Management of security functions behaviour
	FMT_MSA.1a Management of Security Attributes
	FMT_MSA.1b Management of Security Attributes
	FMT_MSA.3a Static Attribute Initialization
	FMT_MSA.3b Static Attribute Initialization
	FMT_REV.1a Revocation
	FMT_SMF.1a Specification of Management Functions
	FMT_SMR.1a Security Management Roles
TOE Protection	FPT_RVM.1a Non-bypassability of the TSP
	FPT_STM.1a Reliable Time Stamps

Table 10 Security Functions vs. Requirements Mapping

8.8 PP Claims Rationale

See Section 7, Protection Profile Claims.