EnterpriseDB

Postgres Plus Advanced Server v8.4

Security Target Version 1.12

June 02, 2011

Prepared For

EnterpriseDB

Prepared By

CYGNACOM SOLUTIONS

7925 Jones Branch Drive♦Suite 5400 ♦McLean, VA 22102-3321♦703 848-0883♦Fax 703 848-0960

TABLE OF CONTENTS

Section **Page**

Table of Tables

Table **Page**

Table of Figures

Figure **Page**

1 Security Target Introduction

1.1 Security Target Reference

ST Title: Enterprise DB Postgres Plus Advanced Server v8.4 Security Target

ST Version: Version 1.12

ST Date: June 02, 2011

ST Author: CygnaCom Solutions

1.2 TOE Reference

TOE Identification: Postgres Plus Advanced Server v8.4

TOE Vendor: EnterpriseDB Corporation

1.3 TOE Overview

Postgres Plus Advanced Server 8.4 is a relational database management system based on PostgreSQL, an open source database. In this ST, the TOE is also referred to as the Advanced Server. The TOE includes the Database Server, and tools for clients, developer and administrators.

The TOE provides the following security functionality: security auditing, Discretionary Access Control (DAC), Identification and Authentication (I&A), security management, protection of the TSF, TOE access, and works with the environment to provide trusted channels.

The TOE is being evaluated at assurance level EAL2 augmented by ALC_FLR.2

The TOE is claiming conformance to the *US Government Protection Profile for Database Management Systems in Basic Robustness Environments, Version 1.2, July 25, 2007.* This PP is referred to in this ST as the DBMS PP.

1.3.1 TOE Type

Postgres Plus Advanced Server 8.4 is a relational database management system (RDBMS). The TOE will be evaluated as a distributed DBMS with multiple copies of the DBMS server and workstations.

1.3.2 Operating System Platforms

The TOE server will be evaluated running on the following operating system platforms:

- DBServer platforms[1]:
    - Red Hat Linux Version 5, and
    - Microsoft Windows 2003 Server

[1] *Although PPAS may provide failover/switchover for crossover platforms configuration (Linux ↔ Windows), that configuration is not recommended and it is not supported by EnterpriseDB. Therefore, crossover platform configuration was not included in the CC evaluated configuration and was not tested during this evaluation. The failover/switchover function was only evaluated for the following configurations of the EDB server:*

Linux RH5 <-> Linux RH5 and Windows 2003 <-> Windows 2003.

- Administrative and Development Platform on Microsoft Windows XP,
- Client Application platform on Linux
- Client Application platform on Microsoft Windows XP

The following TOE components are required to be installed in each of clients and Administrative and Development Platform:

- All the connectors (JDBC, ODBC, .NET, OCI, and libpq)
- Postgres Studio and
- EDB*Plus

Any of the clients can be used as the Administrator Workstation, so there is no need for an additional administrator workstation unless operationally desired.

1.4 TOE Description

1.4.1 Terminology

Table 1-1: Advanced Server Terminology and Table 1-2: DBMS PP Terminology describe product specific and DBMS PP terminology, respectively.

Table 1-1: Advanced Server Terminology

Term	Description
Access Privilege	Object security attribute. If the access privileges column (ACL) is empty for a given object, the object has only the default access privileges.
Accessor	Subject accessing a database object. (Used in the description of the DAC implementation)
Authorized User	An entity that has been properly identified and authenticated. These users are considered to be legitimate users of the TOE.
Authorized Administrator or Administrator	The terms “Authorized Administrator” and “Administrator” are used interchangeably. In this ST the term administrator applies to all users who have authorized access to the TSF Data. This includes both users with Advanced Server Roles with privileges that allow TSF Data access through the TOE’s own interfaces and the OS TOE administrator called the “Cluster owner” who has access the TSF Data through operating system interfaces.
Cluster Owner	A user that is created during the installation process that is given ownership permissions of the TOE. This user is maintained by the OS and can only access the TSF data stored at the OS level after being authenticated at the OS level.
Current_user and session_user	The session user is the user that initiated a database connection; it is fixed for the duration of that connection. The current user is the user identifier that is applicable for permission checking. Normally, it is equal to the session user, but it changes during the execution of functions with the attribute security definer. The session user is the “real user” and the current user is the “effective user.”
Database Administrator	Also known as the Database Superuser or the EDB Superuser in Advanced Server. The Superuser only has access to TSF data via TOE interfaces after authentication.
DBServer	The host computer on which the Database Server component is installed.
DBClient	A workstation that is connected to the DBServer by a secure LAN. Authorized users on the DBClient can access the TOE through a Graphical User Interface, a Command Line Interface, and applications that use Client Connectors.

Term	Description
Default Access privilege	See Access Privilege. Default privileges (default ACL) always include all privileges for the object owner, and may include some privileges for PUBLIC, depending on the object type.
Function	A function is a predefined block of statements that a return a value. The returned value can be of composite type or table type. Functions have a single return value, but can have zero or more input parameters. Functions can be invoked with SQL commands, triggers, operators and indexes. Functions can be created using the CREATE FUNCTION SQL command from Postgres Studio in the evaluated configuration.
Package	A package is a named collection of functions, procedures, variables, cursors, and user-defined record types that are referenced using a common qualifier, the package identifier.
Procedure or Stored Procedure	A procedure is a predefined block of statements. Procedures are invoked using the EXECUTE SQL command or may be invoked from within another function or procedure by including the name of the procedure (and argument list). Procedures can have zero or more input parameters and zero or more output parameters. Procedures are created using the CREATE PROCEDURE SQL command from Postgres SQL in the evaluated configuration.
Role or Advanced Server role	A “role” is used in Advanced Server to define individual users as well as groups of users and sets of access privileges. Note that the term “role” is used somewhat differently in Advanced Server than the more common usage where users are granted roles, but are not roles themselves. A user in Advanced Server is a role that is has been granted the LOGIN privilege. User roles can be granted other roles such as groups and sets of privilege as in other DBMSs. The “in roles” role attribute is used to specify the groups of which a user is a member.
Security Invoker/Definer	This terminology is used for procedures, functions, and packages. SECURITY INVOKER indicates that the procedure, function, or package is to be executed with the privileges of the user that calls it. This is the default. SECURITY DEFINER specifies that the procedure, function, or package is to be executed with the privileges of the user that created it.
Stored Procedure Language	The Stored Procedure Language (SPL) is used to define procedures, functions, packages, and triggers. SPL includes SQL statements as well as programming constructs such as IF-THEN-ELSE, WHILE, LOOP, EXIT, and RETURN
Database Superuser	A Database Superuser is the all-powerful (implicitly granted all privileges) administrator in Advanced Server. Also known as the Database Administrator or DBA. The Database Superuser is called the “authorized administrator” in the DBMS PP.
Trigger	A trigger is a predefined block of statements that are executed when a DELETE, INSERT, or UPDATE command is executed on a table. A trigger is an attribute of a table.
User or Advanced Server user	In Advanced Server, the term “user” refers to an entity representing an individual as in many other IT systems. However, a “user” in Advanced Server is implemented as a role that has been granted the LOGIN privilege.
View	A view is a selection of rows and columns from a table or a set of joined tables. A view can be used to limit access to its underlying tables, since a role can be granted permissions on a view, without granting the role permissions to the view’s underlying tables.

Table 1-2: DBMS PP Terminology

Term	Description
Access	Interaction between an entity and an object that results in the flow or modification of data.
Access Control	Security service that controls the use of hardware and software resources and the disclosure and modification of stored or communicated data.
Accountability	Property that allows activities in an IT system to be traced to the entity responsible for the activity.
Administrator	A user who has been specifically granted the authority to manage some portion or all of the TOE and whose actions may affect the TSP. Administrators may possess special privileges that provide capabilities to override portions of the TSP.
Assurance	A measure of confidence that the security features of an IT system are sufficient to enforce its security policy.
Attack	An intentional act attempting to violate the security policy of an IT system.
Authentication	Security measure that verifies a claimed identity.
Authentication data	Information used to verify a claimed identity.
Authorization	Permission, granted by an entity authorized to do so, to perform functions and access data.
Authorized Administrator	The authorized person in contact with the Target of Evaluation who is responsible for maintaining its operational capability.
Authorized user	An authenticated user who may, in accordance with the TSP, perform an operation.
Availability	Timely (according to a defined metric), reliable access to IT resources.
Compromise	Violation of a security policy.
Confidentiality	A security policy pertaining to disclosure of data.
Conformant Product	A Target of Evaluation that satisfied all the functional security requirements in Section 5.1. The requirements in Section 5.2 are satisfied by its IT environment. Furthermore, a conformant TOE satisfies all the TOE security assurance requirements in section 5.3 of this document.
Critical Security Parameters (CSP)	Security-related information (e.g., cryptographic keys, authentication data such as passwords and pins, and cryptographic seeds) appearing in plaintext or otherwise unprotected form and whose disclosure or modification can compromise the security of a cryptographic module or the security of the information protected by the module.
Database Management System (DBMS)	A suite of programs that typically manage large structured sets of persistent data, offering ad hoc query facilities to many users. They are widely used in business applications.
Defense-in-Depth (DID)	A security design strategy whereby layers of protection are utilized to establish an adequate security posture for an IT system.
Discretionary Access Control (DAC)	A means of restricting access to objects based on the identity of subjects and/or groups to which they belong. Those controls are discretionary in the sense that a subject with an access permission is capable of passing that permission (perhaps indirectly) on to any other subject.
Enclave	A collection of entities under the control of a single authority and having a homogeneous security policy. They may be logical, or may be based on physical location and proximity.
Entity	A subject, object, user or another IT device, which interacts with TOE objects, data, or resources.
External IT entity	Any trusted Information Technology (IT) product or system, outside of the TOE, which may, in accordance with the TSP, perform an operation.
Identity	A representation (e.g., a string) uniquely identifying an authorized user, which can either be the full or abbreviated name of that user or a pseudonym.
Integrity	A security policy pertaining to the corruption of data and TSF mechanisms.
Named Object	An object that exhibits all of the following characteristics:

	• The object may be used to transfer information between subjects of differing user and/or group identities within the TSF. • Subjects in the TOE must be able to require a specific instance of the object. • The name used to refer to a specific instance of the object must exist in a context that potentially allows subjects with different user and/or group identities to require the same instance of the object.
Object	An entity within the TSC that contains or receives information and upon which subjects perform operations.
Operating Environment	The total environment in which a TOE operates. It includes the physical facility and any physical, procedural, administrative and personnel controls.
Public Object	An object for which the TSF unconditionally permits all entities "read" access. Only the TSF or authorized administrators may create, delete, or modify the public objects.
Robustness	A characterization of the strength of a security function, mechanism, service or solution, and the assurance (or confidence) that it is implemented and functioning correctly. DoD has three levels of robustness: • Basic: Security services and mechanisms that equate to good commercial practices. • Medium: Security services and mechanisms that provide for layering of additional safeguards above good commercial practices. • High: Security services and mechanisms that provide the most stringent protection and rigorous security countermeasures.
Secure State	Condition in which all TOE security policies are enforced.
Security attributes	TSF data associated with subjects, objects, and users that are used for the enforcement of the TSP.
Security level	The combination of a hierarchical classification and a set of non-hierarchical categories that represent the sensitivity of the information.
Sensitive information	Information that, as determined by a competent authority, must be protected because its unauthorized disclosure, alteration, loss, or destruction will at least cause perceivable damage to someone or something.
Subject	An entity within the TSC that causes operation to be performed.
Threat	Capabilities, intentions and attack methods of adversaries, or any circumstance or event, with the potential to violate the TOE security policy.
Threat Agent	Any human user or Information Technology (IT) product or system, which may attempt to violate the TSP and perform an unauthorized operation with the TOE.
Unauthorized user	A user who may obtain access only to system provided public objects if any exist.
User	Any entity (human user or external IT entity) outside the TOE that interacts with the TOE.
Vulnerability	A weakness that can be exploited to violate the TOE security policy.

1.4.2 Acronyms

Table 1-3: Advanced Server Acronyms and Table 1-4: CC Acronyms define product specific and CC specific acronyms, respectively. The CC acronyms are taken from the DBMS PP.

Table 1-3: Advanced Server Acronyms

Acronym	Definition
API	Application Programming Interface
CLI	Command Line Interface
DBA	Database Administrator
DDL	Data Definition Language
DBMS	Database Management System
DML	Data Manipulation Language
GSSAPI	Generic Security Services Application Program Interface
GUI	Graphical User Interface
HBA	Host-Based Authentication
LDAP	Lightweight Directory Access Protocol
PAM	Pluggable Authentication Modules
OCI	Oracle Call Interface
RDBMS	Relational DBMS
RMI	Remote Method Invocation
SPL	Stored Procedure Language
SQL	Structured Query Language
SSL	Secure Socket Layer protocol
SSPI	Security Services Provider Interface

Table 1-4: CC Acronyms

Acronym	Definition
CC	Common Criteria
CCIMB	Common Criteria Interpretations Management Board
CCEVS	Common Criteria Evaluation and Validation Scheme
CM	Configuration Management
DAC	Discretionary Access Control
DoD	Department of Defense
EAL	Evaluation Assurance Level
I&A	Identification and Authentication
IAD	NSA Information Assurance Directorate
IATF	Information Assurance Technical Framework
IT	Information Technology
NIAP	National Information Assurance Partnership
NIST	National Institute of Standards and Technology
NSA	National Security Agency
OSP	Organizational Security Policy
PP	Protection Profile
SAR	Security Assurance Requirement
SFP	Security Functional Policy
SFR	Security Functional Requirement
ST	Security Target
TOE	Target of Evaluation
TSC	TOE Scope of Control
TSE	TOE Security Environment
TSF	TOE Security Functionality

Acronym	Definition
TSFI	TSF Interface
TSP	TOE Security Policy

1.4.3 TOE Description

The Postgres Plus Advanced Server (PPAS) is a software-only TOE. The product is made up of the following software components:

1. Database Server 8.4.4-400 (in TOE),
2. Client Connectors (bundled) (in TOE),
3. Postgres Studio 1.10.4 (in TOE),
4. PostGIS Spatial Extensions 1.5.1-3 (in TOE),
5. EDB*Plus 8.4 (build 25) (in TOE),
6. Slony Replication 2.0.3 (in TOE),
7. PG Agent (bundled) (in TOE),
8. Update Monitor (bundled) (in TOE),
9. Infinite Cache Daemon (not in TOE),
10. Migration Studio (not in TOE),
11. EnterpriseDB Migration Toolkit (not in TOE),
12. xDB Replication Server (not in TOE),
13. DBA Management Server (not in TOE),
14. Monitoring Tools (not in TOE),
15. PG Bouncer (not in TOE), and
16. Procedural Language Debugger (not in TOE)
17. StackBuilder Plus (not in TOE)

.

The PPAS Installer displays the above list of product components at installation time. By default, the installer selects the components included in the scope of the evaluation. The components listed as “not in TOE” are prohibited in the scope of the evaluated configuration. Each of the listed components is described in the sections below. A rationale is also provided for those components not in the TOE.

1.4.3.1 Database Server (in TOE)

The database server or DB Server is the relational database engine at the core of the Postgres Plus Advanced Server database server. EnterpriseDB Corporation substantially enhanced PostgreSQL, an open source database, to create the Postgres Plus Advanced Server database server. The Advanced Server database server implements additional named objects such as stored procedures and packages for Oracle compatibility. The database server component provides a Command Line Interface (CLI) that includes: a set of management utilities, the EnterpriseDB Superset Procedural Language (SPL), and the Advanced Server implementation of the SQL language. All of the database server is included in the TOE.

The Advanced Server database server consists of the following subcomponents:

- **PostgreSQL** is an open source database that implements database features such as triggers, functions, and views. The Advanced Server database is built on the PostgreSQL database.

- **Server Utilities** are a collection of command line utilities for managing the database. These commands can only be run usefully on the host where the database server resides.

- **Database Utilities** allow for the creation and removal of databases and database user accounts and retrieving information about the installed version. These command line utilities can be run from a terminal emulation program on any host, independent of where the database server resides.

- **Authentication Support.** The Advanced Server Database Server provides support for multiple authentication mechanisms. Please see Section 1.4.8.3 Identification and Authentication for more information.

- **SQL/Protect:** Module to protect against specific common SQL injection attacks**.**

- **EDB-PSQL:** Command line interface to Database Server

1.4.3.2 Client Connectors (in TOE)

Client Connectors are standardized programming interfaces allow a software developer to connect a customer-specific application to the Advanced Server database. Advanced Server provides connectors for the following enterprise programming environments:

- Java Database Connectivity (JDBC)
- Open Data Base Connectivity (ODBC)
- Microsoft .NET framework
- Libpq, API for client applications written in C
- EnterpriseDB Advanced Server Open Client Library (OCI)

1.4.3.3 Postgres Studio (in TOE)

Postgres Studio is a DBA console and an enterprise-wide, cross-platform development tool. Postgres Studio users can:

- Execute SQL commands
- Use pre-configured wizards for security, backup, and restore
- Browse multiple databases simultaneously
- Conduct SQL query profiling and analysis

Postgres Studio provides a Graphical User Interface (GUI) for its users.

1.4.3.4 PostGIS Spatial Extensions (in TOE)

Post GIS, an open source geographic information server, is built into Advanced Server. PostGIS spatially enables Advanced Server, allowing it to be used as a backend spatial database for geographic information systems (GIS). This is a non-security related component that is contained in the TOE.

1.4.3.5 EDB*Plus (in TOE)

EDB*Plus is a command line interface that offers compatibility with Oracle's SQL Plus commands.

1.4.3.6 Slony Replication (in TOE)

Slony Replication provides database replication services between nodes in a cluster. Slony Replication is as a master-subscriber system that includes the capabilities needed to replicate large databases to a limited number (on the order of a dozen) of subscriber systems. Slony-I implements the model of asynchronous replication, using triggers to collect table updates, where a single “origin” may be replicated to multiple “subscribers” including cascaded subscribers.

1.4.3.7 PG Agent (in TOE)

pgAgent is a job scheduling agent for Postgres, capable of running multi-step batch/shell and SQL tasks on complex schedules.

Note that the default installation of the evaluated configuration of the TOE limits access to the pgAgent tables to Database Superusers; if a Database Superuser should change the privileges associated with the pgAgent tables (granting privileges on the pgAgent tables to a non-Superuser), it may be possible for a non-Superuser to schedule jobs that exceed their designated privileges.

1.4.3.8 Update Monitor (in TOE)

The Update Monitor utility polls the Enterprise DB website and alerts server users (with access to the Postgres Task Manager icon) to security updates and enhancements as they become available for Advanced Server 8.4. This functionality is considered to be non-security related as it is only a notification tool and cannot modify the TOE in any way and does not support any of the identified SFRs in this document.

1.4.3.8.1 StackBuilder Plus (not in TOE)

If an update becomes available, the user has the option to open StackBuilder Plus to download and install the component update.

It should be noted that to install patches/upgrades StackBuilder Plus does require Database Superuser privileges. However, it should also be noted that a Database Superuser can install items that are not included in the evaluated configurations using StackBuilder Plus. Therefore, StackBuilder Plus is not part of the evaluated configuration.

1.4.3.9 Infinite Cache Daemon (not in TOE)

Infinite Cache is a distributed memory caching system that PPAS includes as part of a standard installation on a Linux system. The Advanced Server installation wizard can optionally install only the Infinite Cache daemon on supporting cache servers without installing the Database Server.

With Infinite Cache, PPAS dedicates a portion of the memory installed on each *cache server* as a secondary memory cache. When a client application sends a query to the server, the server first searches the shared buffer cache for the required data; if the requested data is not found in the cache, the server searches for the necessary page in one of the cache servers.

1.4.3.10 Migration Studio (not in TOE)

Migration Studio is a collection of utilities that automatically migrates the data and business logic contained in existing Oracle, Sybase, and MySQL databases to Advanced Server databases.

Use of Migration Studio is prohibited in the evaluated configuration. The rationale for not including it in the scope of the evaluation is that it is a development tool and requires licenses from the other products in order to use it.

1.4.3.11 EnterpriseDB Migration Toolkit (not in TOE)

Migration Toolkit is a command line migration utility that facilitates migration from other database products.

Use of EnterpriseDB Migration Toolkit is prohibited in the evaluated configuration. The rationale for not including it in the scope of the evaluation is that it is a development tool and requires licenses from the other products in order to use it.

1.4.3.12 xDB Replication (not in TOE)

xDB Replication is for replicating data between Oracle and Postgres Plus Advanced Server

Use of xDB Replication is prohibited in the evaluated configuration. The rationale for not including it in the evaluated configuration is that it requires an Oracle license to run it.

1.4.3.13 DBA Management Server (not in TOE)

The Advanced Server DBA Management Server is a database monitoring, profiling, reporting, and querying tool that enables DBAs and developers to analyze, manage and tune multiple Advanced Server or PostgreSQL databases from a single Web browser. It enables database administrators to view audit log files, view database server log files, and view and update runtime configuration parameters.

Use of the DBA Management Server is prohibited in the evaluated configuration. The rationale for not including it in the scope of the evaluation is that it is being deprecated in PPAS release 9.0.

1.4.3.14 Monitoring Tools (not in TOE)

Monitoring tools provide information about CPU, Memory, Disk, and Cache utilization.

Use of the monitoring tools is prohibited in the evaluated configuration. They were not included in the scope of the evaluation, because they are being deprecated in Release 9.0.

1.4.3.15 PG Bouncer (not in TOE)

PgBouncer is a lightweight connection pooling utility for Advanced Server. Connection pooling can dramatically reduce processing time and resources for systems maintaining client connections to one or more databases.

Use of PgBouncer is prohibited in the evaluated configuration. The rationale for not including it in the scope of the evaluated configuration is that it is not intended for use in a production mode.

1.4.3.16 Procedural Language Debugger (not in TOE)

Advanced Server includes the EnterpriseDB Procedural Language (PL) Debugger, used to develop and analyze stored procedures, functions and triggers in applications.

Use of the Procedural Language Debugger is prohibited in the evaluated configuration. This component is not included in the scope of the evaluation, since it is a development tool that EnterpriseDB recommends turning off in production mode.

1.4.4 Users

The users supported by the TOE are the same as those defined in the DBMS PP. The DBMS PP text is copied below:

"A DBMS supports two major types of users:

- *Users who interact with the DBMS to observe and/or modify data objects for which they have authorization to access; and*

- *Authorized administrators who implement and manage the various information-related policies of an organization (e.g., access, integrity, consistency, availability) on the databases that they manage and/or own."*

A "role" is used in Advanced Server to define individual users as well as for groups of users and sets of access privileges. A user in Advanced Server is a role that is has been granted the LOGIN privilege. User roles can be granted other roles such as groups and sets of privilege as in other DBMSs.

In EDB there are two types of authorized administrators. There is the "Database Superuser", also known as the "EDB Superuser", which is a role maintained by the TOE. The "Database Superuser" administers the TOE through the TOE's user interfaces and is the focus of the security functional requirements (SFR)'s described in this document. There is also the "Cluster owner" which is created during the installation of the TOE and is maintained by the OS. The Cluster owner has the OS permissions to modify configuration files stored at the OS level and execute command line interfaces.

1.4.5 Data

The data maintained by the TOE is the same as the definition of DBMS data in the DBMS PP that is copied below:

"A DBMS, in conjunction with the IT environment, stores, and controls access to, two types of data:

- *The first type is the user data that the DBMS maintains and protects. User data may consist of the following:*

  a) *The user data stored in or as database objects;*
  b) *The definitions of user databases and database objects, commonly known as DBMS metadata; and*
  c) *User-developed queries, functions, or procedures that the DBMS maintains for users.*

- *The second type is the DBMS data (e.g., configuration parameters, user security attributes, transaction log, audit instructions and records) that the DBMS maintains and uses to operate the DBMS."*

1.4.6 Product Guidance

The following product guidance documents are provided with the TOE. The documents are available to download from the product web-site.

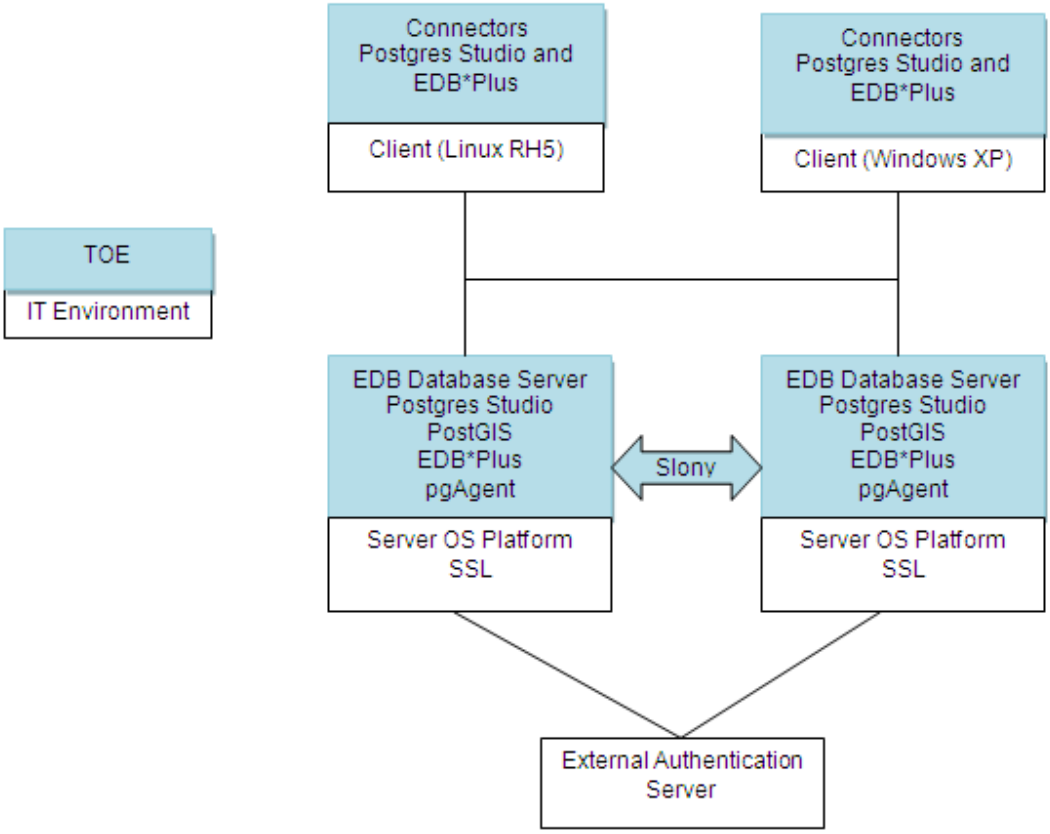
Table 1-5: User Guidance Documents

Reference Title	ID
The PostgreSQL Global Development Group; PostgreSQL 8.4.4 Documentation, Version 8.4.4	PG Online Ref
EnterpriseDB Corp, Postgres Plus® Advanced Server Guide, Version 2.0, June 23, 2010	PPAS
EnterpriseDB Corp, Postgres Plus Advanced Server Oracle Compatibility Developer's Guide, Version 2.17, August 6, 2010	Oracompat
EnterpriseDB Corp, Postgres Plus Advanced Server Postgres Studio Users Guide, Version 1.0., August 8, 2010	PgStud
EnterpriseDB Corp, Postgres Plus Advanced Server Installation Guide; Version 1.0, August 4, 2010	Inst
EnterpriseDB Corp, Postgres Plus Advanced Server 8.4 ODBC Connector Guide; Version 1.1, June 26, 2010	ODBC
The PostgreSQL Global Development Group; PostgreSQL 8.4.4 Documentation, Chapter 30, Version 8.4.4	libpq
EnterpriseDB Corp, Postgres Plus Advanced Server 8.4 JDBC Connector Guide; Version 1.2, June 26, 2010	JDBC
EnterpriseDB Corp, Postgres Plus Advanced Server 8.4 .NET Connector Guide; Version 1.2, August 8, 2010	.NET
EnterpriseDB Corp, Postgres Plus Advanced Server 8.4 Performance Features Guide; Version 1.1, June 27, 2010	ICache
EnterpriseDB Corp, Tutorial: How to Set Up pgAgent for Postgres Plus; Version 1, February 19, 2010	pgAgent
EnterpriseDB Corp, Tutorial: How to Set Up Slony-I Replication for Postgres Plus; Version 1, February 11, 2010	Slony tut
EnterpriseDB Corp, Tutorial: How to use PostGIS with Postgres Plus Advanced Server; Version 2, April 12, 2010	PostGIS tut
Refractions Research, Inc., PostGIS 1.5.1 Manual; Version 1.5.1.	PostGIS

1.4.7 Physical Scope of the TOE

Figure 1-1 below shows a sample configuration with two copies of the Database Server. The figure depicts the physical scope of the TOE within its IT environment.

Figure 1-1: Postgres Plus Advanced Server TOE Boundary



1.4.7.1 In Scope

The following PPAS 8.4 product components are in scope:

- Database Server
- Connectors
- Postgres Studio
- PostGIS Spatial Extensions
- EDB*Plus
- Slony Replication
- PG Agent
- Update Monitor

1.4.7.2 Out of Scope

The following components of the PPAS 8.4 product are not included in the TOE:

- Infinite Cache Daemon
- Migration Studio
- EnterpriseDB Migration Toolkit
- xDB Replication
- DBA Management Server
- Monitoring Tools
- PG Bouncer
- Procedural Language Debugger
- StackBuilder Plus

1.4.7.3 Configuration Options that are Out of Scope.

1.4.7.3.1 "Trust" authentication option (not in TOE)

When the trust authentication option is specified, PostgreSQL assumes that anyone who can connect to the server is authorized to access the database with whatever database user name they specify (including Database Superusers).The use of the EnterpriseDB "trust" authentication option is prohibited in the evaluated configuration, since it configures the TOE to not require any authentication functionality.

1.4.7.3.2 "Ident" authentication option (not in TOE)

The "Identification Protocol" is described in RFC 1413. This authentication method is only appropriate for closed networks where each client machine is under tight control and where the database and system administrators operate in close contact. In other words, the system administrators must trust the machine running the ident server. RFC 1413 issues the following warning: *The Identification Protocol is not intended as an authorization or access control protocol.* Therefore, the use of the "Ident" authentication option is prohibited in the evaluated configuration.

1.4.7.4 IT Environment

No operating systems or platforms are included in the TOE.

In addition, the following components in the IT environment are out of scope.

- Authenticator servers, if configured
- Terminal emulator
- Syslog server, if configured (not tested as there is no security requirements tied to this interface)
- SNMP server, if configured (not tested as there is no security requirements tied to this interface)

1.4.7.4.1 Functional Dependencies on the IT Environment

Advanced Server relies on the IT environment for the following security functionality:

- Storage of audit records in operating system files
- Text Viewer to review audit records
- Identification and Authentication methods that rely upon authentication servers and/or operating system platforms in the IT environment (PAM, LDAP, Kerberos, GSSAPI, SSPI, SSL Certs)
- Identification and Authentication of the “Cluster owner” OS user
- Maintenance of Cluster owner’s password and security attributes
- Storage of the TOE configuration files
- Text Editor to edit the TOE’s configuration files stored at the OS level
- Reliable timestamps from the OS
- OS protection of TOE programs and data (audit, configuration files, executables, and db)
- SSL on the Database Server platform (OpenSSL 0.9.8) and the client and administrator workstations

1.4.8 Logical Scope of the TOE

1.4.8.1 Security Audit

Advanced Server generates audit records for security relevant events. The TOE provides the capability to select auditable events based on settings in a system configuration files.

1.4.8.2 User Data Protection

Advanced Server provides Discretionary Access Control (DAC) that controls access to objects based on the identity of the subjects or groups to which the subjects and objects belong. The TOE allows authorized users to specify how the objects that they control are protected. The TOE provides the capability to grant privileges (e.g., SELECT, INSERT, UPDATE, DELETE,TRUNCATE, CREATE, EXECUTE, and USAGE) on relational database objects such as tables, columns, views, triggers, functions, procedures, tablespaces and schemas. These privileges can be granted to roles. (Note that in Advanced Server, a role with the LOGIN privilege is used for an individual user.) The TOE also provides for the inheritance of privileges between roles. Explicit delegation of privileges on a database object among users is also permitted.

1.4.8.3 Identification and Authentication

Advanced Server ensures that users are identified and authenticated by some method before allowing access to TSF resources. The available methods (*auth-method: parameter*) for client authentication definition include:

- Password (*password*)

- MD5 Password (*md5*)
- Pluggable Authentication Modules (*pam*)
- Lightweight Directory Access Protocol (*ldap*)
- Kerberos (*krb5*)
- Generic Security Services API (*gss*)
- Security Service Provider Interface (*sspi*)
- SSL Certificates (*cert*)

Password and MD5 Password functionality is completely provided by the TOE. The other authentication methods require the support of authentication servers and/or operating systems in the IT environment. Note that the use of the "Trust" or "Ident" authentication methods are prohibited in the evaluated configuration.

One additional authentication method parameter identified in the guidance documentation, but is not covered in this section, is called *reject*. This parameter is used to explicitly deny session establishment and is included in the Section 1.4.8.6 TOE Access description. The *reject* authentication mechanism option does not provide any means of successful I&A.

Note: The MD5 implementation is vendor developed to the RFC 1321 specification and is strictly used for password hashing. The MD5 cryptographic function implementation, or module, has not been FIPS certified. The correctness of the cryptographic module used by the TOE is by Vendor assertion; the correctness and conformance of this cryptographic module to the RFC 1321 standard is not be part of this evaluation.

1.4.8.4 Security Management

Advanced Server provides security management through the server command line utilities, database command line utilities, Postgres Studio, and the DBA Management Server.

The TOE provides an authorized administration role (Database Superuser) to allow authorized administrators to perform security management functions. Users with the CREATEDB and CREATEROLE privileges are also trusted administrative roles in Advanced Server.

Security management also includes the ability to revoke user and object security attributes.

1.4.8.5 Protection of the TSF

The TOE provides a way to replicate changes to data on one database server to the other database servers within a cluster. The TOE provides the functionality to switchover or failover from the master database server to a replicated database server upon the request of the Database Superuser. The Cluster owner is responsible for setting the persistent parameters in the pg_hba.conf configuration file stored at the OS level. Additional parameters can be modified by the Database Superuser.

The TOE also provides protection against SQL injection attacks by examining incoming queries for common SQL injection attacks such as unbounded DML statements, unauthorized relations, SQL tautology, and utility commands.

1.4.8.6 TOE Access

Advanced Server is able to restrict the maximum number of concurrent sessions that belong to the same user.

PPAS provides users with the ability to view their own connection history based on information recorded in the audit log. Users can retrieve information about connection history. The history includes a list of connection attempts with a date and time stamp of each connection, and a determination whether the connection was successful and unsuccessful thus allowing the user to determine the number of unsuccessful attempts since the last successful session establishment.

The TSF can deny session establishment based on user identity, group identity, database name, Host IP address, and/or subnet address, and the *maximum number of connections allowed to the server* threshold. The functionality to deny a session based on user identity, group identity, database name, Host IP address, and/or subnet address is tied into the authentication mechanism functionality, as described in Section 1.4.3.8 Identification and Authentication, using the *auth-method: parameter* called *reject.* The *maximum number of connections allowed to the server* threshold is a global server setting.

1.4.8.7 Partial Trusted Communication

The TOE works in conjunction with the IT environment to provide trusted communication between the DB Server and Postgres Studio and between DB Server and clients in the IT environment using SSL.

2 Conformance Claims

2.1 Common Criteria Conformance

The TOE is Part 2 extended, Part 3 conformant, and meets the requirements of Evaluation Assurance Level (EAL) 2 augmented by ALC_FLR.2 Flaw reporting procedures from Common Criteria Version 3.1 R2.

2.2 Protection Profile Claim

The TOE claims demonstrable conformance to the *US Government Protection Profile for Database Management Systems in Basic Robustness Environments, Version 1.2, July 25, 2007.* This PP is referred to in this ST as the DBMS PP.

Demonstrable conformance is defined in Part 1 of CC v3.1 r2 as follows:

> *"There is no subset-superset type relation between the PP and the ST. The PP and the ST may contain entirely different statements that discuss different entities, use different concepts etc. However, the ST shall contain a rationale on why the ST is considered to be "equivalent or more restrictive" than the PP (see Section D.3). Demonstrable conformance allows a PP author to describe a common security problem to be solved and provide generic guidelines to the requirements necessary for its resolution, in the knowledge that there is likely to be more than one way of specifying a resolution. Demonstrable conformance is also suitable for a TOE type where several similar PPs already exist (or likely to exist in the future), thus allowing the ST author to claim conformance to all these PPs simultaneously, thereby saving work. (paragraph 1)"*

Paragraph 445 provides additional details on what is required for *demonstrable* compliance in the areas of:

- Security problem definition
- Security objectives
- Security requirements

2.2.1 Security Problem Definition

CC Text

The conformance rationale in the ST shall demonstrate that the security problem definition in the ST is equivalent (or more restrictive) than the security problem definition in the PP. This means that:

- All TOEs that would meet the security problem definition in the ST also meet the security problem definition in the PP;
- All operational environments that would meet the security problem definition in the PP would also meet the security problem definition in the ST.

Conformance rationale:

This Security Target includes all of the threats, organizational security policies, and assumption statements described in the PP, verbatim.

In addition, the ST includes T.DENIAL_OF_SERVICE, because the TOE includes controlled switchover and failover functionality to mitigate the threat of denial of service, if the master database server does down.

Therefore the security problem definition in this ST is equivalent or more restrictive than the security problem definition in the PP.

2.2.2 Security Objectives

CC Text

The conformance rationale in the ST shall demonstrate that the security objectives in the ST is equivalent (or more restrictive) than the security objectives in the PP. This means that:

- All TOEs that would meet the security objectives for the TOE in the ST also meet the security objectives for the TOE in the PP;
- All operational environments that would meet the security objectives for the operational environment in the PP would also meet the security objectives for the operational environment in the ST.

Conformance Rationale

This Security Target includes all of the TOE Security Objectives from the PP. In addition, ST also includes the security objectives O. AUTH, OE.AUTH, O.AVAIL, and OE.PROTCOMM.

O.AUTH was included, because the TOE provides its own identification and authentication mechanisms. In addition, OE.AUTH was included, because the TOE supports authentication by authentication servers and operating systems in the IT environment.

O.AVAIL was included, because the TOE provides controlled switchover and failover functionality to support availability.

O.PROTCOMM was included, because the TOE provides SSL between the DB Server and Postgres Studio and between the DB Server and its clients in the IT environment.

OE.PROTCOMM was included, because the TOE relies upon SSL running clients. SSL is used for session establishment and to protect TSF data from unauthorized disclosure or modification when it is transmitted between distributed parts of the TOE. The ST objectives are still equivalent to the DBMS PP objectives, because it does not result in any TOE requirements being moved to the IT environment and is equivalent to relying upon the OS in the IT environment.

The security objectives in this ST are therefore equivalent to or more restrictive than the security objectives in the PP.

2.2.3 Security Requirements

CC Text

The ST shall contain all SFRs and SARs in the PP, but may claim additional or hierarchically stronger SFRs and SARs. The completion of operations in the ST must be consistent with that in the PP; either

the same completion will be used in the ST as that in the PP or one that makes the requirement more restrictive (the rules of refinement apply).

Conformance Rationale
This Security Target includes all of the Security Functional Requirements and Security Assurance Requirements from the PP. The completed operations of the ST are consistent with those in the PP. The Security Target also includes the following additional Security Functional Requirements:

- FIA_UAU_(EXT).2 Partial authentication before any other TSF-mediated action
- FIA_UAU.5 Multiple authentication mechanisms
- FIA_UAU_(EXT).5 Partial multiple authentication mechanisms
- FIA_UID_(EXT).2 Partial identification before any other TSF-mediated action
- FPT_OVR_(EXT).1 Database server switchover / failover
- FPT_SIP_(EXT).1 Partial SQL injection protection
- FTP_ITC_(EXT).1 Partial trusted channels

2.3 Package Claim

The TOE meets the requirements for Evaluation Assurance Level (EAL2) augmented by ALC_FLR.2 Flaw reporting procedures.

3 Security Problem Definition

The security environment for the functions addressed by this specification includes threats, security policies, and usage assumptions, as discussed below.

3.1 Threats

A Basic Robustness TOE is considered sufficient for low threat environments or where compromise of protected information will not have a significant impact on mission objectives. This implies that the motivation of the threat agents will be low in environments that are suitable for TOEs of this robustness. In general, basic robustness results in “good commercial practices” that counter threats based in casual and accidental disclosure or compromise of data protected by the TOE.

The TOE must counter the threats to security listed in Table 3-1. All the threats from the DBMS PP are included as well as T.DENIAL_OF_SERVICE.

Table 3-1: TOE Threats

Item	Threat ID	Threat Description
1	T. ACCIDENTAL_ADMIN_ ERROR	An administrator may incorrectly install or configure the TOE resulting in ineffective security mechanisms.
2	T.MASQUERADE	A user or process may masquerade as another entity in order to gain unauthorized access to data or TOE resources
3	T.POOR_DESIGN	Unintentional errors in requirements specification or design of the TOE may occur, leading to flaws that may be exploited by a casually mischievous user or program.
4	T.POOR_IMPLEMENTATION	Unintentional errors in implementation of the TOE design may occur, leading to flaws that may be exploited by a casually mischievous user or program.
5	T.POOR_TEST	Lack of or insufficient tests to demonstrate that all TOE security functions operate correctly (including in a fielded TOE) may result in incorrect TOE behavior being discovered thereby causing potential security vulnerabilities.
6	T.RESIDUAL_DATA	A user or process may gain unauthorized access to data through reallocation of TOE resources from one user or process to another.
7	T.TSF_COMPROMISE	A malicious user or process may cause configuration data to be inappropriately accessed (viewed, modified or deleted).
8	T.UNAUTHORIZED_ACCESS	A user may gain unauthorized access to user data for which they are not authorized according to the TOE security policy.
9	T.UNIDENTIFIED_ACTIONS	Failure of the authorized administrator to identify and act upon unauthorized actions may occur.
10	T.DENIAL_OF_SERVICE	Failure of the master database server might cause the database to become unavailable to users.

3.2 Organizational Security Policies

The Organizational Security Policies of the TOE are defined in Table 3-2. Both of these organizational security policies come from the DBMS PP.

Table 3-2: TOE Organizational Security Policies

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

3.3 Assumptions

The assumptions regarding the security environment and the intended usage of the TOE are listed in Table 3-3. All these assumptions come from the DBMS PP.

Table 3-3: Assumptions

Item	Assumption ID	Assumption Description
1	A.NO_EVIL	Administrators are non-hostile, appropriately trained, and follow all administrator guidance.
2	A.NO_GENERAL_PURPOSE	There are no general-purpose computing capabilities (e.g., compilers or user applications) available on DBMS servers, other than those services necessary for the operation, administration and support of the DBMS.
3	A.OS_PP_VALIDATED	The underlying OS has been validated against an NSA sponsored OS PP of at least Basic Robustness.
4	A.PHYSICAL	It is assumed that appropriate physical security is provided within the domain for the value of the IT assets protected by the TOE and the value of the stored, processed, and transmitted information.

4 Security Objectives

4.1 Security Objectives for the TOE

The security objectives for the TOE are listed in Table 4-1. The last column indicates whether or not the objective came from the DBMS PP.

Table 4-1: TOE Security Objectives

Item	TOE Objective	Description	From DBMS PP
1	O.ACCESS_HISTORY	The TOE will store and retrieve information (to authorized users) related to previous attempts to establish a session.	Yes
2	O.ADMIN_GUIDANCE	The TOE will provide administrators with the necessary information for secure management.	Yes
3	O.ADMIN_ROLE	The TOE will provide authorized administrator roles to isolate administrative actions.	Yes
4	O.AUDIT_GENERATION	The TOE will provide the capability to detect and create records of security relevant events associated with users.	Yes
5	O.CONFIGURATION_IDENTIFICATION	The configuration of the TOE is fully identified in a manner that will allow implementation errors to be identified and corrected with the TOE being redistributed promptly.	Yes
6	O.DOCUMENTED_DESIGN	The design of the TOE is adequately and accurately documented.	Yes
7	O.INTERNAL_TOE_DOMAINS	The TSF will maintain internal domains for separation of data and queries belonging to concurrent users.	Yes
8	O.MANAGE	The TOE will provide all the functions and facilities necessary to support the authorized administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.	Yes
9	O.MEDIATE	The TOE must protect user data in accordance with its security policy.	Yes
10	O.PARTIAL_FUNCTIONAL_TEST	The TOE will undergo some security functional testing that demonstrates that the TSF satisfies some of its security functional requirements.	Yes
11	O.PARTIAL_SELF_PROTECTION	The TSF will maintain a domain for its own execution that protects itself and its resources from external interference, tampering, or unauthorized disclosure through its own interfaces.	Yes
12	O.RESIDUAL_INFORMATION	The TOE will ensure that any information contained in a protected resource within its Scope of Control is not released when the resource is reallocated.	Yes
13	O.TOE_ACCESS	The TOE will provide mechanisms that control a user's logical access to the TOE.	Yes

Item	TOE Objective	Description	From DBMS PP
14	O.VULNERABILITY_ANALYSIS	The TOE will undergo some vulnerability analysis to demonstrate that the design and implementation of the TOE does not contain any obvious flaws.	Yes
15	O. AUTH	The TSF will ensure that all users are identified and authenticated before allowing them to access to TSF-mediated resources. The TSF with the support of the IT environment will support multiple authentication methods.	No
16	O.AVAIL	The TOE will provide controlled switchover and failover capabilities to increase the availability of the database.	No
17	O.PROTCOMM	The TOE will protect communication between the DB Server and Postgres Studio and between DB Server and clients in the IT environment using SSL.	No

4.2 Security Objectives for the Operational Environment

The security objectives for the Operational Environment are listed in Table 4-2.

Table 4-2: Security Objectives for the Operational Environment

Item	Environment Objective	Description	From DBMS PP
1	OE.NO_EVIL	Sites using the TOE shall ensure that authorized administrators are non-hostile, appropriately trained and follow all administrator guidance.	Yes
2	OE.NO_GENERAL_ PURPOSE	There will be no general-purpose computing capabilities (e.g., compilers or user applications) available on DMBS servers, other than those services necessary for the operation, administration and support of the DBMS.	Yes
3	OE.OS_PP_VALIDATED	The underlying OS has been validated against an NSA sponsored OS PP of at least Basic Robustness.	Yes
4	OE.PHYSICAL	Physical security will be provided within the domain for the value of the IT assets protected by the TOE and the value of the stored, processed, and transmitted information.	Yes
5	OE.AUTH	The IT environment will provide support for the authentication mechanisms invoked by the TSF,	No
6	OE.PROTCOMM	SSL running on clients will support protection TSF data when it is transmitted between them and the DB Server	No

4.3 Security Objectives Rationale

Table 4-3: Mapping of TOE Security Objectives to Threats/Policies

Item	TOE Objective	Threat
1	O.ACCESS_HISTORY	T.UNAUTHORIZED_ACCESS

Item	TOE Objective	Threat
2	O.ADMIN_GUIDANCE	T. ACCIDENTAL_ADMIN_ERROR T.UNIDENTIFIED_ACTIONS
3	O.ADMIN_ROLE	P.ROLES
4	O.AUDIT_GENERATION	P.ACCOUNTABILITY
5	O.CONFIGURATION_IDENTIFICATION	T.POOR_DESIGN T.POOR_IMPLEMENTATION
6	O.DOCUMENTED_DESIGN	T.POOR_DESIGN T.POOR_TEST
7	O.INTERNAL_TOE_DOMAINS	T.TSF_COMPROMISE
8	O.MANAGE	T.TSF_COMPROMISE T.UNIDENTIFIED_ACTIONS
9	O.MEDIATE	T.UNAUTHORIZED_ACCESS
10	O.PARTIAL_FUNCTIONAL_TEST	T.POOR_IMPLEMENTATION T.POOR_TEST
11	O.PARTIAL_SELF_PROTECTION	T.TSF_COMPROMISE
12	O.RESIDUAL_INFORMATION	T.RESIDUAL_DATA T.TSF_COMPROMISE
13	O.TOE_ACCESS	P.ACCOUNTABILITY T.MASQUERADE
14	O.VULNERABILITY_ANALYSIS	T.POOR_DESIGN T.POOR_IMPLEMENTATION T.POOR_TEST
15	O.AUTH	P.ACCOUNTABILITY
16	O.AVAIL	T.DENIAL_OF_SERVICE
17	O.PROTCOMM	T.TSF_COMPROMISE

Table 4-4: Mapping of Security Objectives for the Operational Environment to Threats/Policies/Assumptions

Item	Environment Objective	Threat/Policy/Assumption
1	OE.NO_EVIL	A.NO_EVIL
2	OE.NO_GENERAL_PURPOSE	A.NO_GENERAL_PURPOSE
3	OE.OS_PP_VALIDATED	A.OS_PP_VALIDATED
4	OE.PHYSICAL	A.PHYSICAL
5	OE.AUTH	P.ACCOUNTABILITY
6	OE.PROTCOMM	T.TSF_COMPROMISE

Table 4-5 shows that all the identified Threats to security are countered by Security Objectives. Rationale is provided for each Threat in the table.

Table 4-5: All Threats to Security Countered

#	Threat ID	Objective	Rationale

#	Threat ID	Objective	Rationale
1	T.ACCIDENTAL_ADMIN_ERROR An administrator may incorrectly install or configure the TOE resulting in ineffective security mechanisms.	O.ADMIN_GUIDANCE The TOE will provide administrators with the necessary information for secure management.	O.ADMIN_GUIDANCE helps to mitigate this threat by ensuring the TOE administrators have guidance that instructs them how to administer the TOE in a secure manner. Having this guidance helps to reduce the mistakes that an administrator might make that could cause the TOE to be configured in insecurely.
2	T.MASQUERADE A user or process may masquerade as another entity in order to gain unauthorized access to data or TOE resources.	O.TOE_ACCESS The TOE will provide mechanisms that control a user's logical access to the TOE.	O.TOE_ACCESS mitigates this threat by controlling the logical access to the TOE and its resources. By constraining how and when authorized users can access the TOE, and by mandating the type and strength of the authentication mechanism this objective helps mitigate the possibility of a user attempting to login and masquerade as an authorized user. In addition, this objective provides the administrator the means to control the number of failed login attempts a user can generate before an account is locked out, further reducing the possibility of a user gaining unauthorized access to the TOE.
3	T.POOR_DESIGN Unintentional errors in requirements specification or design of the TOE may occur, leading to flaws that may be exploited by a casually mischievous user or program.	O.CONFIGURATION_IDENTIFICATION The configuration of the TOE is fully identified in a manner that will allow implementation errors to be identified and corrected with the TOE being redistributed promptly.	O.CONFIGURATION_IDENTIFICATION plays a role in countering this threat by requiring the developer to provide control of the changes made to the TOE's design.
		O.DOCUMENTED_DESIGN The design of the TOE is adequately and accurately documented.	O.DOCUMENTED_DESIGN ensures that the design of the TOE is documented, permitting detailed review by evaluators.
		O.VULNERABILITY_ANALYSIS The TOE will undergo some vulnerability analysis to demonstrate the design and implementation of the TOE does not contain any obvious flaws.	O.VULNERABILITY_ANALYSIS ensures that the design of the TOE is analyzed for design flaws.
4	T.POOR_IMPLEMENTATION Unintentional errors in implementation of the TOE design may occur, leading to flaws that may be exploited by a casually mischievous user or program.	O.CONFIGURATION_IDENTIFICATION The configuration of the TOE is fully identified in a manner that will allow implementation errors to be identified and corrected with the TOE being redistributed promptly.	O.CONFIGURATION_IDENTIFICATION plays a role in countering this threat by requiring the developer to provide control of the changes made to the TOE's design, although the previous three objectives help minimize the introduction of errors into the implementation.

#	Threat ID	Objective	Rationale
		O.PARTIAL_FUNCTIONAL_TEST The TOE will undergo some security functional testing that demonstrates the TSF satisfies some of its security functional requirements.	O.PARTIAL_FUNCTIONAL_TEST increases the likelihood that any errors that do exist in the implementation (with respect to the functional specification, high level, and low-level design) will be discovered through testing.
		O.VULNERABILITY_ANALYSIS The TOE will undergo some vulnerability analysis to demonstrate the design and implementation of the TOE does not contain any obvious flaws.	O.VULNERABILITY_ANALYSIS helps reduce errors in the implementation that may not be discovered during functional testing. Ambiguous design documentation and the fact that exhaustive testing of the external interfaces is not required may leave bugs in the implementation undiscovered in functional testing.
5	T.POOR_TEST Lack of or insufficient tests to demonstrate that all TOE security functions operate correctly (including in a fielded TOE) may result in incorrect TOE behavior being discovered thereby causing potential security vulnerabilities.	O.DOCUMENTED_DESIGN The design of the TOE is adequately and accurately documented.	O.DOCUMENTED_DESIGN helps to ensure that the TOE's documented design satisfies the security functional requirements. In order to ensure the TOE's design is correctly realized in its implementation, the appropriate level of functional testing of the TOE's security mechanisms must be performed during the evaluation of the TOE.
		O.PARTIAL_FUNCTIONAL_TEST The TOE will undergo some security functional testing that demonstrates the TSF satisfies some of its security functional requirements.	O.PARTIAL_FUNCTIONAL_TEST increases the likelihood that any errors that do exist in the implementation (with respect to the functional specification, high level, and low-level design) will be discovered through testing.

#	Threat ID	Objective	Rationale
		O.VULNERABILITY_ANALYSIS The TOE will undergo some vulnerability analysis to demonstrate the design and implementation of the TOE does not contain any obvious flaws.	O.VULNERABILITY_ANALYSIS addresses this concern by requiring a vulnerability analysis be performed in conjunction with testing that goes beyond functional testing. This objective provides a measure of confidence that the TOE does not contain security flaws that may not be identified through functional testing. While these testing activities are a necessary activity for successful completion of an evaluation, this testing activity does not address the concern that the TOE continues to operate correctly and enforce its security policies once it has been fielded. Some level of testing must be available to end users to ensure the TOE's security mechanisms continue to operator correctly once the TOE is fielded.
6	T.RESIDUAL_DATA A user or process may gain unauthorized access to data through reallocation of TOE resources from one user or process to another.	O.RESIDUAL_INFORMATION The TOE will ensure that any information contained in a protected resource within its Scope of Control is not released when the resource is reallocated.	O.RESIDUAL_INFORMATION counters this threat by ensuring that TSF data and user data is not persistent when resources are released by one user/process and allocated to another user/process.
7	T.TSF_COMPROMISE A user or process may cause, through an unsophisticated attack, TSF data, or executable code to be inappropriately accessed (viewed, modified, or deleted).	O.RESIDUAL_INFORMATION The TOE will ensure that any information contained in a protected resource within its Scope of Control is not released when the resource is reallocated.	O.RESIDUAL_INFORMATION is necessary to mitigate this threat, because even if the security mechanisms do not allow a user to view TSF data, if TSF data were to reside inappropriately in a resource that was made available to a user, that user would be able to view the TSF data without authorization.
		O.PARTIAL_SELF_PROTECTION The TSF will maintain a domain for its own execution that protects itself and its resources from external interference, tampering, or unauthorized disclosure through its own interfaces.	O.PARTIAL_SELF_PROTECTION ensures the TOE is capable of protecting itself from attack.
		O.MANAGE The TOE will provide all the functions and facilities necessary to support the authorized administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.	O.MANAGE is necessary because an access control policy is specified to control access to TSF data. This objective is used to dictate who is able to view and modify TSF data, as well as the behavior of TSF functions.

#	Threat ID	Objective	Rationale
		O.INTERNAL_TOE_DOMAINS The TSF will maintain internal domains for separation of data and queries belonging to concurrent users.	O.INTERNAL_TOE_DOMAINS ensures the TOE will establish separate domains for
		O.PROTCOMM The TOE will protect communication between the DB Server and Postgres Studio and between DB Server and clients in the IT environment.	O.PROTCOMM ensures that the DB Server and Postgres Studio provides SSL for the protection communication between the DB Server and Postgres Studio and the DB server and clients in the IT environment.
		OE.PROTCOMM SSL running on clients will support protection TSF data when it is transmitted between them and the DB Server.	OE.PROTCOMM ensures that SSL on clients in the IT environment will support protection of TSF data when it is transmitted between DB Server and clients.
8	T.UNAUTHORIZED_ACCESS A user may gain unauthorized access to user data for which they are not authorized according to the TOE security policy.	O.MEDIATE The TOE must protect user data in accordance with its security policy.	O.MEDIATE ensures that all accesses to user data are subject to mediation, unless said data has been specifically identifies as public data. The TOE requires successful authentication to the TOE prior to gaining access to any controlled-access content. By implementing strong authentication to gain access to these services, an attacker's opportunity to conduct a man-in-the-middle and/or password guessing attack successfully is greatly reduced. Lastly, the TSF will ensure that all configured enforcement functions (authentication, access control rules, etc.) must be invoked prior to allowing a user to gain access to TOE or TOE mediated services. The TOE restricts the ability to modify the security attributes associated with access control rules, access to authenticated and unauthenticated services, etc to the administrator. This feature ensures that no other user can modify the information flow policy to bypass the intended TOE security policy.
		O.ACCESS_HISTORY The TOE will store and retrieve information (to authorized users) related to previous attempts to establish a session.	O.ACCESS_HISTORY is important to mitigate this threat because it ensures the TOE will be able to store and retrieve the information that will advise the user of the last successful login attempt and performed actions without their knowledge.

#	Threat ID	Objective	Rationale
9	T.UNIDENTIFIED_ACTIONS Failure of the authorized administrator to identify and act upon unauthorized actions may occur.	O.ADMIN_GUIDANCE The TOE will provide administrators with the necessary information for secure management.	The threat of an authorized administrator failing to know about malicious audit events produces the objectives of the authorized administrator having the facilities and knowing how to use them (O.ADMIN_GUIDANCE).
		O.MANAGE The TOE will provide all the functions and facilities necessary to support the authorized administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.	The threat of an authorized administrator failing to know about malicious audit events produces the objectives of the authorized administrator having the capability to use the mechanisms (O.MANAGE) to review audit records.
10	T.DENIAL_OF_SERVICE	O.AVAIL The TOE will provide controlled switchover and failover capabilities to increase the availability of the database.	The threat of a failure of the master database server making data unavailable to users is mitigated by the TOE providing controlled switchover and failover capabilities

Table 4-6 shows that the security objectives for the operational environment enforce all Organizational Security Policies. Rationale is provided for each Policy in the table.

Table 4-6: All Security Policies Enforced

OSP ID	Objective	Rationale
P.ACCOUNTABILITY The authorized users of the TOE shall be held accountable for their actions within the TOE.	O.AUDIT_GENERATION The TOE will provide the capability to detect and create records of security relevant events associated with users.	O.AUDIT_GENERATION addresses this policy by providing the authorized administrator with the capability of configuring the audit mechanism to record the actions of a specific user, or review the audit trail based on the identity of the user. Additionally, the administrator's ID is recorded when any security relevant change is made to the TOE (e.g., access rule modification, start-stop of the audit mechanism, establishment of a trusted channel, etc.).
	O. AUTH The TSF will ensure that all users are identified and authenticated before allowing them to access to TSF-mediated resources. The TSF with the support of the IT environment will support multiple authentication methods.	O.AUTH supports this policy by requiring the TOE to identify and authenticate all authorized users prior to allowing any TOE access or any TOE mediated access on behalf of those users.
	O.TOE_ACCESS The TOE will provide mechanisms that control a user's logical access to the TOE.	O.TOE_ACCESS supports this policy by requiring the TOE to identify and authenticate all authorized users prior to allowing any TOE access or any TOE mediated access on behalf of those users.

OSP ID	Objective	Rationale
	OE.AUTH The IT environment will provide support for the authentication mechanisms invoked by the TSF,	OE.AUTH supports this policy by requiring the IT environment to provide authentication services to support identification and authentication of all authorized users.
P.ROLES The TOE shall provide an authorized administrator role for secure administration of the TOE. This role shall be separate and distinct from other authorized users.	O.ADMIN_ROLE The TOE will provide authorized administrator roles to isolate administrative actions.	The TOE has the objective of providing an authorized administrator role for secure administration. The TOE may provide other roles as well, but only the role of authorized administrator is required

Table 4-7 shows that the security objectives for the operational environment uphold all assumptions. Rationale is provided for each Assumption in the table.

Table 4-7: All Assumptions Upheld

#	Assumption ID	Objective	Rationale
1	A.NO_EVIL Administrators are non-hostile, appropriately trained, and follow all administrator guidance.	OE.NO_EVIL Sites using the TOE shall ensure that authorized administrators are non-hostile, are appropriately trained and follow all administrator guidance.	All authorized administrators are trustworthy individuals, having background investigations commensurate with the level of data being protected, have undergone appropriate admin training, and follow all admin guidance.
2	A.NO_GENERAL_PURPOSE There are no general-purpose computing or storage repository capabilities (e.g., compilers or user applications) available on DBMS servers, other than those services necessary for the operation, administration and support of the DBMS.	OE.NO_GENERAL_PURPOSE There will be no general-purpose computing capabilities (e.g., compilers or user applications) available on DMBS servers, other than those services necessary for the operation, administration and support of the DBMS.	The DBMS server must not include any general-purpose commuting or storage capabilities. This will protect the TSF data from malicious processes.
3	A.OS_PP_VALIDATED It is assumed that the underlying OS has been validated against an NSA sponsored OS PP of at least Basic Robustness.	OE.OS_PP_VALIDATED	The underlying OS must be validated to at least basic robustness to ensure it provides an appropriate level of protection for the DBMS. The OS must provide domain separation, Non-bypassability, Audit Review, Audit Storage, Identification and Authentication.

#	Assumption ID	Objective	Rationale
4	A.PHYSICAL Physical security, commensurate with the value of the TOE and the data it contains, is assumed to be provided by the IT environment.	OE.PHYSICAL Physical security will be provided within the domain for the value of the IT assets protected by the TOE and the value of the stored, processed, and transmitted information.	The TOE, the TSF data, and protected user data is assumed to be protected from physical attack (e.g., theft, modification, destruction, or eavesdropping). Physical attack could include unauthorized intruders into the TOE environment, but it does not include physical destructive actions that might be taken by an individual that is authorized to access the TOE environment.

5 Extended Components Definition

All of the components defined below have been modeled on components from Part 2 of the CC Version 3.1. The extended components are denoted by adding "_(EXT)" in the component name.

5.1 Extended Component Rationale from DBMS PP

Table 5-1: Extended Component Rationale from the DBMS PP

Component	Component Title	Rationale from DBMS PP
FAU_GEN_(EXT).2	User and/or group identity association	This requirement was needed to replace FAU_GEN.2.1-NIAP-0410 because this PP does not require the TOE to implement a user identity. It does require the TOE to implement a user identity and/or a group identity to satisfy the DAC policy. Therefore, this extended requirement was created to allow the audit function to use the user identity or the group identity or both.
FPT_TRC_(EXT).1	Internal TSF consistency	FPT_TRC_(EXT).1 has been created to require timely consistency of replicated TSF data. Although there is a Common Criteria Requirement that attempts to address this functionality, it falls short of the needs of the environment in this protection profile. Specifically, FPT_TRC.1.1 states "The TSF shall ensure that TSF data is consistent when replicated between parts of the TOE." In the widely distributed environment of this PP's TOE, this is an infeasible requirement. For TOEs with a very large number of components, 100 percent TSF data consistency is not achievable and is not expected at any specific instant in time. Another concern lies in FPT_TRC.1.2 that states that when replicated parts of the TSF are "disconnected", the TSF shall ensure consistency of the TSF replicated data upon "reconnection". Upon first inspection, this seems reasonable, however, when applying this requirement it becomes clear that it dictates specific mechanisms to determine when a component is "disconnected" from the rest of the TSF and when it is "reconnected". This is problematic in this PP's environment in that it is not the intent of the authors to dictate that distributed TSF components keep track of connected/disconnected components. In general, to meet the needs of this PP, it is acceptable to only require a mechanism that provides TSF data consistency in a timely manner after it is determined that it is inconsistent.
FTA_TAH_(EXT).1	TOE Access History	This PP does not require the TOE to contain a client. Therefore, the PP cannot require the client to display a message. This requirement has been modified to require the TOE to store and retrieve the access history instead of displaying it.
FMT_MSA_(EXT).3	Static attribute initialization	The CC does not allow the PP author to specify restrictive values that are not modifiable. This extended requirement eliminates the element FMT_MSA.3.2 from the component FMT_MSA.3 and makes the component more secure by requiring the security attributes of the objects on creation to be restrictive and not allowing any user to be able of override the restrictive default values.
FIT_PPC_(EXT).1	IT Environment Protection Profile Compliance	This requirement is necessary to ensure the TOE will be running on an OS that is at least as robust as the TOE itself.

5.2 Extended Component Definition for Components not Drawn from DBMS PP

Table 5-2 below lists the extended components defined by the ST author in the sections below.

Table 5-2: Extended Components Defined by the ST Author

Component	Component Name
FIA_UAU_(EXT).2	Partial authentication before any other TSF-mediated action
FIA_UAU_(EXT).5	Partial multiple authentication mechanisms
FIA_UID_(EXT).2	Partial identification before any other TSF-mediated action
FPT_OVR_(EXT).1	Partial internal data transfer protection
FPT_SIP_(EXT).1	Partial SQL Injection Protection
FTP_ITC_(EXT).1	Partial trusted channels

5.2.1 FIA_UAU_(EXT).2 Partial authentication before any other TSF-mediated action

5.2.1.1 Class

FIA: Identification and Authentication

5.2.1.2 Family

FIA_UAU: User authentication

5.2.1.3 Family Behaviour

This family defines the types of user authentication mechanisms supported by the TSF. This family also defines the required attributes on which the user authentication mechanisms must be based.

5.2.1.4 Management

Management of the authentication data by an administrator
Management of the authentication data by the user associated with this data

5.2.1.5 Audit

Minimal: Unsuccessful use of the authentication mechanism

5.2.1.6 Definition

Hierarchical to: No other components
Dependencies: FIA_UID_(EXT).2

FIA_UAU_(EXT).2.1 The TSF shall require each user or group to be successfully authenticated with the support of the IT environment before allowing any other TSF-mediated actions on behalf of that user or group.

5.2.1.7 Rationale

FIA_UAU_(EXT).2 is modeled on the Part 2 component: FIA_UAU.2.

This component needs to be defined as an extended component, because the TOE requires the TSF and IT Environment working together to provide authentication of all users or groups before allowing them to perform any TSF-mediated action.

5.2.2 FIA_UAU_(EXT).5 Partial multiple authentication mechanisms

5.2.2.1 Class

FIA: Identification and Authentication

5.2.2.2 Family

FIA_UAU: User authentication

5.2.2.3 Family Behaviour

This family defines the types of user authentication mechanisms supported by the TSF. This family also defines the required attributes on which the user authentication mechanisms must be based.

5.2.2.4 Management

Management of authentication mechanisms

5.2.2.5 Audit

Minimal: The final decision on authentication

5.2.2.6 Definition

Hierarchical to: No other components.
Dependencies: No dependencies.

FIA_UAU_(EXT).5.1 The TSF with the support of the IT environment shall provide the following authentication mechanisms:

- Pluggable Authentication Modules (PAM)
- Lightweight Directory Access Protocol (LDAP)
- Kerberos
- Generic Security Services API (GSSAPI)
- Security Service Provider Interface (SSPI)
- SSL Certificates

to support user and group authentication.

FIA_UAU_(EXT).5.2 The TSF with the support of the IT environment shall authenticate any user's claimed identity using the authentication mechanism configured by the authorized administrator.

5.2.2.7 Rationale

FIA_UAU_(EXT).5 is modeled on the Part 2 component FIA_UAU.5.

This component needed to be defined as an extended component because the TOE requires the TSF and IT Environment working together to support the multiple authentication methods that may be invoked by the TSF.

5.2.3 FIA_UID_(EXT).2 Partial identification before any other TSF-mediated action

5.2.3.1 Class

FIA: Identification and Authentication

5.2.3.2 Family

FIA_UID: User identification

5.2.3.3 Family Behaviour

This family defines the conditions under which users shall be required to identify themselves before performing any other actions that are to be mediated by the TSF and which required user identification

5.2.3.4 Management

Management of the user identities

5.2.3.5 Audit

Minimal: Unsuccessful use of the identification mechanism, including the identity provided;

5.2.3.6 Definition

Hierarchical to: No other components
Dependencies: No dependencies.

FIA_UID_(EXT).2.1 The TSF with the support of the IT environment shall require each user or group to be successfully identified before allowing any other TSF-mediated actions on behalf of that user or group.

5.2.3.7 Rationale

FIA_UID_(EXT).2 is modeled on the Part 2 component FIA_UID.2

This component needs to be defined as an extended component, because the TOE requires the TSF and IT Environment working together to provide identification of all users before allowing them to perform any TSF-mediated action.

5.2.4 FPT_OVR_(EXT).1 Database Server Switchover/Failover

5.2.4.1 Class

FPT: Protection of the TSF

5.2.4.2 Family

FPT_OVR: TSF Switchover/Failover

5.2.4.3 Family Behaviour

This family provides requirements that address the switchover or failover of one TSF component to another TSF component.

5.2.4.4 Management

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

a) management of the controlled switchover function

b) management of the failover function

5.2.4.5 Audit

a) initiation of controlled switchover

b) completion of controlled switchover

c) initiation of failover

d) completion of failover

5.2.4.6 Definition

Database Server Controlled Switchover/Failover (FPT_OVR_(EXT).1)

Hierarchical to: No other components.
Dependencies: FPT_TRC_(EXT).1 Internal TSF consistency

FPT_OVR_(EXT).1 The TSF shall provide the capability to switchover the master node from the current master node to a subscriber node, upon an authorized administrator issued MOVE SET command, with no loss of transactions if both nodes are operational at the time of the switchover.

FPT_OVR_(EXT).2 The TSF shall provide the capability to failover from a master node to a subscriber node, upon an authorized administrator issued FAILOVER command, with only the loss of transactions that have been committed on the master node, but not on the subscriber node.

Application note: The subscriber node is a replicated copy of the master node.

5.2.4.7 Rationale

This component is loosely modeled on FPT_ITA.1
This component had to be explicitly stated, because there is no existing CC component to specify the functionality of controlled switchover or failover. FPT_ITA.1 addresses the availability of exported data, but it requires that availability be specified in terms of “an identified degree of probability”. FPT_OVR_EXT.1 specifies functionality that is dependent upon the timeliness of the internal TSF consistency function.

5.2.5 FPT_SIP_(EXT).1 Partial SQL Injection Protection

5.2.5.1 Class

FPT: Protection of the TSF

5.2.5.2 Family

FPT_SIP: SQL Injection Attack Protection

5.2.5.3 Family Behaviour

This family provides requirements that address the TOE detecting SQL injection attacks against itself and taking action as configured by the administrator.

5.2.5.4 Management

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

a) Configuration of the SQL Injection Attack Protection function

5.2.5.5 Audit

a) Detection of a SQL injection attempt

5.2.5.6 Definition

Partial SQL Injection Protection (FPT_SIP_(EXT).1)

Hierarchical to: No other components.
Dependencies: None

FPT_SIP_(EXT).1 The TSF shall be able to detect the following types of SQL Injection Attacks using signatures:

- Unauthorized Relations
- Utility Commands
- SQL Tautology
- Unbounded DML

FPT_SIP_(EXT).2 The TSF shall be able to record and display potential SQL injection attacks for review by the authorized administrator.

FPT_SIP_(EXT).3 The TSF shall have the capability block SQL commands that have been identified as potential SQL injection attacks.

5.2.5.7 Rationale

This component is loosely modeled after IDS components in the IDS family in the IDS protection profiles.

FPT_SIP_(EXT).1 is modeled after IDS_SDC.1.1 and IDS_ANL.1.1
FPT_SIP_(EXT).2 is modeled after IDS_SDC.1.2 and IDS_ANL.1.2
FPT_SIP_(EXT).3 is modeled after IDS_RCT.1.1

This component had to be explicitly stated, because there is no existing CC component to specify the functionality of provided by the TOE. The SQL Injection Attack protection functionality provided by the TOE protects the TOE itself rather than a third party system, so it has been included in the FPT class. No one IDS components encompassed the capability that the TOE provides.

5.2.6 FTP_ITC_(EXT).1 Partial trusted channels

5.2.6.1 Family: Inter-TSF trusted channel (FTP_ITC)

See Section 18.1 of the Common Criteria for Information Technology Security Evaluation Part 2: Security functional components July 2009 Version 3.1 Revision 3 for the behavior of the FDP_ITC family.

5.2.6.2 Management

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

- Configuring the actions that require trusted channel, if supported.

5.2.6.3 Audit

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

- Minimal: Failure of the trusted channel functions.
- Minimal: Identification of the initiator and target of failed trusted channel functions.
- Basic: All attempted uses of the trusted channel functions.
- Basic: Identification of the initiator and target of all trusted channel functions.

5.2.6.4 Definition

FTP_ITC_(EXT).1 Partial trusted channels
Hierarchical to: No other components.
Dependencies: No dependencies.
FTP_ITC_(EXT).1.1 The Database Server shall provide an encrypted communication channel between itself and Postgres Studio and between itself and trusted clients that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure using SSL..
FTP_ITC_(EXT).1.2 The Database Server shall permit Postgres Studio and the clients in the IT Environment entities to initiate communication via the trusted channel.

5.2.6.5 Rationale

FTP_ITC_(EXT).1 needed to be extended because the TOE provides both internal trusted channels between some of its components and external trusted channels to trusted components in the IT environment.. Also, although SSL support is compiled into the Database Server and SSL is configured and managed through the TOE, the TOE relies upon OpenSSL on the underlying platform to perform the cryptographic operations.

6 Security Requirements

This section provides the security functional and assurance requirements for the TOE.

The notation, formatting, and conventions used in this security target (ST) are consistent with version 3.1 of the Common Criteria for Information Technology Security Evaluation. The CC permits operations to be performed on functional and assurance components when specifying the security requirements for the TOE. In this Security Target, operations are only performed on the security functional components.

Table 6-1: Formatting Conventions for Requirement Operations

Operation	Purpose of Operation	Formatting Convention	Format Example
Refinement by the PP author	The **refinement** operation is used to add detail to a requirement, and thus further restricts a requirement	Refinement of security requirements is denoted by "**Refinement**" in **bold text** following the SFR ID. The added text is also in **bold text**. The deleted text is provided in Appendix A: Text Omitted from DBMS PP	**Refinement: Added Text in Bold**
Selection by the PP author	The **selection** operation is used to select one or more options provided by the CC in stating a requirement.	Selections that have been made by the PP author are denoted by *italic text*	*Italic*
Selection by the ST author	The **selection** operation is used to select one or more options provided by the CC in stating a requirement.	Selections completed by the ST author appear in ***bold italic*** text inside square brackets.	***[bracketed bold italic]***
Assignment by the PP author	The **assignment** operation is used to assign a specific value to an unspecified parameter, such as the length of a password.	Assignments that have been made by the PP authors are denoted by showing the value *text* inside square brackets,	[bracketed text]
Assignment by the ST author	The **assignment** operation is used to assign a specific value to an unspecified parameter, such as the length of a password.	Assignments completed by the ST author appear in bold text inside square brackets.	**[bracketed bold text]**
Iteration	The **iteration** operation is used when a component is repeated with varying operations.	Short name followed by iteration number in parentheses	FMT_REV.1(2)
Extended Requirement	**Extended requirements** created by PP and ST authors instead of copied from the CC. Their use is permitted, if the CC does not offer suitable requirements to meet the authors' needs. They must be defined before they are used. This is done in Section 5.	Family name followed by **_(EXT)** followed by component number.	FAU_GEN_(EXT).2
NIAP Interpretation	**NIAP interpretations** were used in the DBMS PP.	NIAP interpretations are presented with the NIAP interpretation number as part of the requirement identifier	FAU_SEL.1-NIAP-0407
Interpretation	**Interpretation Notes** are provided	Interp Note: followed by colon in	Italic Times New

Operation	Purpose of Operation	Formatting Convention	Format Example
note by PP author	to show the reader where international interpretations have modified a requirement.	*italic Times New Roman* font. These modifications will be displayed before or after the affected element.	Roman
Application note by PP author	**Application Notes** are provided to help the developer, either to clarify the intent of a requirement, identify implementation choices, or to define "pass-fail" criteria for a requirement.	Application Note: followed by colon in *italic Arial* font. The Application Notes will follow the requirement component.	Italic Arial
Application note by ST author	**Application Notes** are provided to help the developer, either to clarify the intent of a requirement, identify implementation choices, or to define "pass-fail" criteria for a requirement.	Application Note: followed by colon in *italic Courier New font*	Italic Courier New

6.1 Security Functional Requirements for the TOE

The functional security requirements for the TOE are listed in Table 6-2: Functional Components below.

They are taken from either the DBMS PP or are extended components defined in Section 5.

Table 6-2: Functional Components

#	SFR Short Name	SFR Description	From DBMS PP	Extended	Refined
1	FAU_GEN.1-NIAP-0410	Audit data generation	yes	no	yes
2	FAU_GEN_(EXT).2	User and/or group identity association	yes	yes	no
3	FAU_SEL.1-NIAP-0407	Selective audit	yes	no	yes
4	FDP_ACC.1	Subset access control	yes	no	no
5	FDP_ACF.1-NIAP-0407	Security attribute based access control	yes	no	yes
6	FDP_RIP.1	Subset residual information protection	yes	no	no
7	FIA_ATD.1	User attribute definition	yes	no	no
8	FIA_UAU_(EXT).2	Partial authentication before any other TSF-mediated action	no	yes	no
9	FIA_UAU.5	Multiple authentication mechanisms	no	no	no
10	FIA_UAU_(EXT).5	Partial multiple authentication mechanisms	no	yes	no
11	FIA_UID_(EXT).2	Partial identification before any other TSF-mediated action	no	yes	no
12	FMT_MOF.1	Management of security functions behavior	yes	no	no
13	FMT_MSA.1	Management of security attributes	yes	no	yes
14	FMT_MSA_(EXT).3	Static attribute initialization	yes	yes	no
15	FMT_MTD.1	Management of TSF data	yes	no	no
16	FMT_REV.1(1)	Revocation (user attributes)	yes	no	no

#	SFR Short Name	SFR Description	From DBMS PP	Extended	Refined
17	FMT_REV.1(2)	Revocation (subject, object attributes)	yes	no	no
18	FMT_SMF.1	Specification of management functions	yes	no	no
19	FMT_SMR.1	Security roles	yes	no	yes
20	FPT_ OVR_(EXT).1	Database server switchover/failover	no	yes	no
21	FPT_SIP_(EXT).1	Partial SQL Injection Protection	no	yes	no
22	FPT_TRC_(EXT).1	Internal TSF consistency	yes	yes	no
23	FTA_MCS.1	Basic limitation on multiple concurrent sessions	yes	no	yes
24	FTA_TAH_(EXT).1	TOE access history	yes	yes	no
25	FTA_TSE.1	TOE session establishment	yes	no	yes
26	FTP_ITC_(EXT).1	Partial Trusted Channels	no	yes	no

6.1.1 Security Audit (FAU)

6.1.1.1 Audit data generation (FAU_GEN.1-NIAP-0410)

Hierarchical to: No other components.
Dependencies: FPT_STM.1 Reliable time stamps

FAU_GEN.1.1-NIAP-0410 **Refinement**: The TSF shall be able to generate an audit record of the following auditable events:

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

b) All auditable events for the *minimum* level of audit **listed in Table 6-3**

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

d) Use of special permissions (e.g., those often used by authorized administrators to circumvent access control policies); and

e) ***[events commensurate with a minimal level of audit introduced by the inclusion of extended requirements determined by the ST author listed in Table 6-4.]***

FAU_GEN.1.2-NIAP-0410 The TSF shall record within each audit record at least the following information:

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

b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [information specified in column three of Table 6-3 and Table 6-4 below].

Table 6-3: Auditable Events from DBMS PP

Security Functional Requirement	Auditable Event(s)	Additional Audit Record Contents
FAU_GEN.1-NIAP-0410	None	
FAU_GEN_(EXT).2	None	
FAU_SEL.1-NIAP-0407	All modifications to the audit configuration that occur while the audit collection functions are operating	The identity of the authorized administrator that made the change to the audit configuration
FDP_ACC.1	None	
FDP_ACF.1-NIAP-0407	Successful requests to perform an operation on an object covered by the SFP	The identity of the subject performing the operation
FDP_RIP.1	None	
FIA_ATD.1	None	
FMT_MOF.1	None	
FMT_MSA.1	None	
FMT_MSA_(EXT).3	None	
FMT_MTD.1	None	
FMT_REV.1(1)	Unsuccessful revocation of security attributes	Identity of individual attempting to revoke security attributes
FMT_REV.1(2)	Unsuccessful revocation of security attributes	Identity of individual attempting to revoke security attributes
FMT_SMF.1	Use of the management functions	Identity of the administrator performing these functions
FMT_SMR.1	Modifications to the group of users that are part of a role	Identity of authorized administrator modifying the role definition
FPT_TRC_(EXT).1	Restoring consistency	
FTA_MCS.1	Rejection of a new session based on the limitation of multiple concurrent sessions	
FTA_TAH_(EXT).1	None	
FTA_TSE.1	Denial of a session establishment due to the session establishment mechanism	Identity of the individual attempting to establish a session

Table 6-4: Auditable Events for Additional Requirements

Security Functional Requirement	Auditable Event(s)	Additional Audit Record Contents
FIA_UAU_(EXT).2	Unsuccessful use of the authentication mechanism	Identity provided
FIA_UAU.5	The final decision on authentication	Identity provided
FIA_UAU_(EXT).5	The final decision on authentication	Identity provided
FIA_UID_(EXT).2	Unsuccessful use of the user identification mechanism	Identity provided
FPT_OVR_(EXT).1	Initiation of controlled switchover Completion of controlled switchover Initiation of failover Completion of failover	
FPT_SIP_(EXT.1)	Refusal/Denial of illegal utility Refusal/Denial of illegal query Refusal/Denial of illegal command type	Identity provided
FTP_ITC_(EXT).1	None	

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

Hierarchical to: No other components.
Dependencies:
FAU_GEN.1 Audit data generation
FIA_UID.1 Timing of identification

FAU_GEN_(EXT).2.1 For audit events resulting from actions of identified users and/or identified groups, the TSF shall be able to associate each auditable event with the identity of the user and/or group that caused the event.

Application Note: A user in Advanced Server is a role with the LOGIN privilege. The "in roles" attribute is used to specify the groups of which a user is a member. (Please refer to FIA_ATD.1 for a complete list of role/user/group attributes)

6.1.1.3 Selective audit (FAU_SEL.1-NIAP-0407)

Hierarchical to: No other components.
Dependencies:
FAU_GEN.1 Audit data generation
FMT_MTD.1 Management of TSF data

FAU_SEL.1.1-NIAP-0407 **Refinement**: The TSF shall allow only the administrator to include or exclude auditable events from the set of audited events based on the following attributes:

a) *user identity and/or group identity*,
b) event type,
c) *object identity,*
d)[success of auditable security events;
e) failure of auditable security events; and
f) [**severity level**], and
g) **[audit record fields listed in Table 6-3 and Table 6-4 above]].**

6.1.2 User data protection (FDP)

6.1.2.1 Subset access control (FDP_ACC.1)

Hierarchical to: No other components.
Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACC.1.1 The TSF shall enforce the [Discretionary Access Control policy] on [all subjects, all DBMS-controlled objects and all operations among them].

6.1.2.2 Security attribute based access control (FDP_ACF.1-NIAP-0407)

Hierarchical to: No other components.
Dependencies:
FDP_ACC.1 Subset access control
FMT_MSA.3 Static attribute initialization

FDP_ACF.1.1-NIAP-0407 The TSF shall enforce the [Discretionary Access Control policy] to objects based on the following:

- [Authorized user identity and/or group membership associated with a subject;
- Access operations implemented for DBMS-controlled objects; and
- Object identity].

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

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

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

FDP_ACF.1.3-NIAP-0407 **Refinement:** The TSF shall explicitly authorize access of subjects to **DBMS-controlled** objects based on the following additional rules: **[subject has authorized administrator role].**

FDP_ACF.1.4-NIAP-0407 The TSF shall explicitly deny access of subjects to objects based on the following rules: ***[no additional explicit denial rules].***

6.1.2.3 Subset residual information protection (FDP_RIP.1)

Hierarchical to: No other components.
Dependencies: No dependencies

FDP_RIP.1.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon the allocation of the resource to [**database objects listed under FDP_ACC.1**].

6.1.3 Identification and authentication (FIA)

6.1.3.1 User attribute definition (FIA_ATD.1)

Hierarchical to: No other components.
Dependencies: No dependencies.

FIA_ATD.1.1 The TSF shall maintain the following list of security attributes belonging to individual users:

- [Database user identifier and/or group memberships;
- Security-relevant database roles; and
- **[Role security attributes listed in Table 6-5 below]**]

Table 6-5: Advanced Server Role Security Attributes

Attribute	Default Value	Security Function	Notes
name	none	Identification & Authentication, Access Control	Required field Same as Advanced Server user identity for roles that have LOGIN attribute value
Database Superuser	NOSUPERUSER	Access Control	A role with the SUPERUSER attribute can override all access restrictions. Only a Database Superuser can create a new Database Superuser
createuser	NOCREATEUSER	Access Control	Deprecated. CREATEUSER is an alternate way to specify Database Superuser.
createdb	NOCREATEDB	Access Control	A role with the CREATEDB attribute can create new databases
createrole	NOCREATEROLE	Access Control	A role with the CREATEROLE attribute can create, alter and drop other roles
inherit	INHERIT	Access Control	A role with the inherit attribute can automatically use whatever database privileges have been granted to all roles it is directly or indirectly a member of. Without INHERIT, membership in another role only grants the ability to SET ROLE to that other role; the privileges of the other role are only available after having done so.
login	NOLOGIN	Identification & Authentication	A role with the login attribute is allowed to log in; that is, the role can be given as the initial session authorization name during client connection. A role having the LOGIN attribute value is a user. Roles without this attribute are useful for managing database privileges, but are not users.
connection limit	-1	Identification & Authentication	Specifies how many concurrent connections the role can make. -1 means no limit.
password	none	Identification & Authentication	A password is only of use for roles having the LOGIN attribute value. This attribute is only necessary for password authentication.

Attribute	Default Value	Security Function	Notes
encrypted	default behavior is determined by the configuration parameter password_encryption	Identification & Authentication	Controls whether the password is stored encrypted in the system catalogs. If the presented password string is already in MD5-encrypted format, then it is stored encrypted as-is, regardless of whether ENCRYPTED or UNENCRYPTED is specified (since the system cannot decrypt the specified encrypted password string). This allows reloading of encrypted passwords during dump/restore.
valid until *<date>*	none	Identification & Authentication	Date and time after which the role's password is no longer valid. If it is omitted the password will be valid for all time.
in role	none	Access Control	Specifies one or more existing roles (groups) of which the new role is a member.

6.1.3.2 Partial authentication before any other TSF-mediated action (FIA_UAU_(EXT).2)

Hierarchical to: No other components
Dependencies: FIA_UID_(EXT).2

FIA_UAU_(EXT).2.1 The TSF shall require each user or group to be successfully authenticated with the support of the IT environment before allowing any other TSF-mediated actions on behalf of that user or group.

6.1.3.3 Multiple authentication mechanisms (FIA_UAU.5)

Hierarchical to: No other components.
Dependencies: No dependencies.

FIA_UAU.5.1 The TSF shall provide the following authentication mechanisms:

[
- ***Password (password option)***
- ***MD5 Password (MD5 option)***

]

to support user and group authentication.

FIA_UAU.5.2 The TSF shall authenticate any user's claimed identity according to the [**rules describing how the multiple authentication mechanisms provide authentication**].

6.1.3.4 Partial multiple authentication mechanisms (FIA_UAU_(EXT).5)

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_UAU_(EXT).5.1 The TSF with the support of the IT environment shall provide the following authentication mechanisms:

- **Pluggable Authentication Modules (*pam* option)**
- **Lightweight Directory Access Protocol (*ldap* option)**
- **Kerberos (*krb5* option)**
- **Generic Security Services API (*gss* option)**
- **Security Service Provider Interface (*sspi* option)**
- **SSL Certificates (*cert* option)**

to support user and group authentication.

FIA_UAU_(EXT).5.2 The TSF with the support of the IT environment shall authenticate any user's claimed identity using the authentication mechanism configured by the authorized administrator.

6.1.3.5 Partial identification before any other TSF-mediated action (FIA_UID_(EXT).2)

Hierarchical to: No other components
Dependencies: No dependencies.

FIA_UID_(EXT).2.1 The TSF with the support of the IT environment shall require each user or group to be successfully identified before allowing any other TSF-mediated actions on behalf of that user or group.

6.1.4 Security management (FMT)

6.1.4.1 Management of security functions behavior (FMT_MOF.1)

Hierarchical to: No other components.
Dependencies:
FMT_SMR.1 Security roles
FMT_SMF.1 Specification of Management Functions

FMT_MOF.1.1 The TSF shall restrict the ability to [**perform the operations specified in Table 6-6**] to the [**functions listed in Table 6-6**] to [**the authorised identified roles listed in Table 6-6**].

Table 6-6: Management of Security Functions Behavior

#	Operation	Function	Authorized Roles
1	Determine the behaviour of	Residual information protection function	Database Superuser
2	Determine the behaviour of	Intrusion detection	Database Superuser
3	Determine the behaviour of	Session establishment	Database Superuser
4	Determine the behaviour of	Authentication functions	Database Superuser
5	Determine the behaviour of	User identification function	Database Superuser Role with CREATEROLE security attribute
6	Revoke	User security attributes	Database Superuser

#	Operation	Function	Authorized Roles
			Role with CREATEROLE security attribute
7	Revoke	Object security attributes	Database Superuser
8	View	Database Performance Statistics	Database Superuser
9	View	Database server configuration parameters	Database Superuser
10	View	System Catalog data	Database Superuser
11	Reload	Database server configuration parameters	Database Superuser

Application Notes:

1) The first two rows are derived from the DBMS PP. Some DBMS events such as connect are logged within the OS.

2) The "authorized administrator" is called the "Database Superuser" in Advanced Server.

3) Roles with the CREATEROLE security attribute can CREATE, ALTER, and DROP other roles.

6.1.4.2 Management of security attributes (FMT_MSA.1)

Hierarchical to: No other components.
Dependencies:
FDP_ACC.1 Subset access control
FMT_SMR.1 Security roles
FMT_SMF.1 Specification of Management Functions

FMT_MSA.1.1 **Refinement**: The TSF shall enforce the [Discretionary Access Control policy] to restrict the ability to [manage] all the security attributes to [authorized administrators].

6.1.4.3 Static attribute initialization (FMT_MSA_(EXT).3)

FMT_MSA_(EXT).3.1 The TSF shall enforce the [Discretionary Access Control policy] to provide *restrictive* default values for security attributes that are used to enforce the SFP.

6.1.4.4 Management of TSF data (FMT_MTD.1)

Hierarchical to: No other components.
Dependencies:
FMT_SMR.1 Security roles
FMT_SMF.1 Specification of Management Functions

FMT_MTD.1.1 The TSF shall restrict the ability to [**perform operations specified in Table 6-7**] on the [**TSF data specified in Table 6-7**] to [**the authorised identified roles specified in Table 6-7**].

Table 6-7: Management of TSF Data

#	Operation	TSF Data	Authorized Role
1	Create and delete	Advanced Server database	Database Superuser Role with the CREATEDB attribute

2	Create, delete and modify	Roles	Database Superuser Role with the CREATEROLE attribute value
3	Set (modify)	Maximum allowed number of concurrent user sessions	Database Superuser Role with the CREATEROLE attribute value
4	Set (modify)	Run-time configuration parameters (not persistent beyond server restart)	Database Superuser
5	Set (modify)	Authentication data	Database Superuser Role with the CREATEROLE attribute value
6	Set (modify)	User's own authentication data	User (i.e., role with LOGIN attribute) associated with the data
7	Set (modify)	Replication	Database Superuser
8	Set (modify)	Controlled switchover parameters	Database Superuser
9	Set (modify)	Failover parameters	Database Superuser
10	Import	Bulk Data transfer	Database Superuser

Application Notes:

1) The first two rows are derived from the DBMS PP. Some DBMS events such as connect are logged within the OS.

2) The "authorized administrator" is called the "Database Superuser" in Advanced Server.

3) Some DBMS configuration options are set by the "Cluster owner", a trusted operating system user who has been granted the necessary permissions to modify Advanced Server configuration files. The Cluster owner can also start, stop, and restart the server.

6.1.4.5 Revocation (FMT_REV.1(1)) [Users]

Hierarchical to: No other components.
Dependencies: FMT_SMR.1 Security roles

FMT_REV.1.1(1) The TSF shall restrict the ability to revoke security attributes associated with the users within the TSC to **[the Database Superuser or a role with the CREATEROLE attribute**].

FMT_REV.1.2(1) The TSF shall enforce the rule **[**

- **User privileges are revoked using the REVOKE command.**
- **Both user privileges and user dependent privileges are revoked using the CASCADE option for the REVOKE command.**
- **Users can only revoke the direct privileges that they have granted to another user.**
- **Users can only revoke the indirect privileges that they have granted to another user.**
- **The revocation takes effect at the next user login]**.

6.1.4.6 Revocation (FMT_REV.1(2)) [Objects]

FMT_REV.1.1(2) The TSF shall restrict the ability to revoke security attributes associated with the objects within the TSC to ***[the Database Superuser and database users as allowed by the Discretionary Access Control policy].***

FMT_REV.1.2(2) The TSF shall enforce the following rules: **[**

- **Object privileges are revoked using the REVOKE command.**
- **Both object privileges and object dependent privileges are revoked using the CASCADE option for the REVOKE command.**
- **Users can only revoke the direct privileges that they have granted on an object.**
- **Users can only revoke the indirect privileges that they have granted on an object.**
- **The revocation takes effect the next time that the object is opened.]**

6.1.4.7 Specification of Management Functions (FMT_SMF.1)

Hierarchical to: No other components.
Dependencies: No dependencies

FMT_SMF.1.1 The TSF shall be capable of performing the following security management functions:

- **[Security management functions listed in Table 6-6 (see FMT_MOF.1),**
- **Management of security attributes (see FMT_MSA.1),**
- **Management of TSF data listed in Table 6-7 (see FMT_MTD.1), and**
- **Revocation of user and object security attributes (see FMT_REV.1(*).]**

6.1.4.8 Security roles (FMT_SMR.1)

Hierarchical to: No other components.
Dependencies: FIA_UID.1 Timing of identification

FMT_SMR.1.1 **Refinement**: The TSF shall maintain the roles:

- [Authorized administrator];
- **[Customized roles created by the authorized administrator based on security attributes (SUPERUSER, CREATEROLE, and CREATEDB privileges) defined in FIA_ATD.1]**

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

Application notes:

1) The "authorized administrator" is called the "Database Superuser" in Advanced Server.

6.1.5 Protection of the TOE Security Functions (FPT)

6.1.5.1 SQL Injection Protection (FPT_SIP_(EXT).1)

Hierarchical to: No other components.
Dependencies: None

FPT_SIP_(EXT).1 The TSF shall be able to detect the following types of SQL Injection Attacks using signatures:

- Unauthorized Relations
- Utility Commands
- SQL Tautology
- Unbounded DML

FPT_SIP_(EXT).2 The TSF shall be able to record and display potential SQL injection attacks for review by the authorized administrator.

FPT_SIP_(EXT).3 The TSF shall have the capability block SQL commands that have been identified as potential SQL injection attacks.

6.1.5.2 Database Server Controlled Switchover/Failover (FPT_OVR_(EXT).1)

Hierarchical to: No other components.
Dependencies: FPT_TRC_(EXT).1 Internal TSF consistency

FPT_OVR_(EXT).1 The TSF shall provide the capability to switchover the master node from the current master node to a subscriber node, upon an authorized administrator issued MOVE SET command, with no loss of transactions if both nodes are operational at the time of the switchover.

FPT_OVR_(EXT).2 The TSF shall provide the capability to failover from a master node to a subscriber node, upon an authorized administrator issued FAILOVER command, with only the loss of transactions that have been committed on the master node, but not on the subscriber node.

Application note:
The subscriber node is a replicated copy of the master node.

6.1.5.3 Internal TSF consistency (FPT_TRC_(EXT).1)

Hierarchical to: No other components.
Dependencies: FPT_ITT.1 Basic internal TSF data transfer protection

FPT_TRC_(EXT).1.1 The TSF shall ensure that TSF data is consistent between parts of the TOE by providing a mechanism to bring inconsistent TSF data into a consistent state in a timely manner.

Application note:
The timeliness of replication is controlled by the Cluster owner setting run-time configuration parameters for SLONY-I slon daemon. Please see Section 7.1.5.1 PT-1: Internal TSF Consistency *for more details.*

6.1.6 TOE Access (FTA)

6.1.6.1 Basic limitation on multiple concurrent sessions (FTA_MCS.1)

Hierarchical to: No other components.
Dependencies: FIA_UID.1 Timing of identification

FTA_MCS.1.1 The TSF shall restrict the maximum number of concurrent sessions that belong to the same user.

FTA_MCS.1.2 **Refinement**: The TSF shall enforce, by default, a limit of ***[an administrator configurable number of]*** sessions per user.

6.1.6.2 TOE access history (FTA_TAH_(EXT).1)

Hierarchical to: No other components.
Dependencies: No dependencies.

FTA_TAH_(EXT).1.1 Upon successful session establishment, the TSF shall store and retrieve the *date and time* of the last successful session establishment to the user.

FTA_TAH_(EXT).1.2 Upon successful session establishment, the TSF shall store and retrieve the *date and time* of the last unsuccessful attempt to session establishment and the number of unsuccessful attempts since the last successful session establishment.

6.1.6.3 TOE session establishment (FTA_TSE.1)

Hierarchical to: No other components.
Dependencies: No dependencies.

FTA_TSE.1.1 **Refinement**: The TSF shall be able to deny session establishment based on [attributes that can be set explicitly by authorized administrator(s), including user identity and/or group identity], and ***[database name, Host IP address, and/or subnet address***].

6.1.7 Trusted Path/Channels

6.1.7.1 Partial Trusted Channel (FTP_ITC_(EXT).1)

Hierarchical to: No other components.
Dependencies: No dependencies.
FTP_ITC_(EXT).1.1 The Database Server shall provide an encrypted communication channel between itself and Postgres Studio and between itself and trusted clients that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure using SSL.
FTP_ITC_(EXT).1.2 The Database Server shall permit Postgres Studio and the clients in the IT Environment entities to initiate communication via the trusted channel.

6.2 Security Requirements for the IT Environment

6.2.1 IT Environment (FIT)

6.2.1.1 IT Environment Protection Profile Compliance (FIT_PPC_(EXT).1)

FIT_PPC_(EXT).1.1 The IT environment shall be compliant with the requirements of the Controlled Access Protection Profile or an Operating System Protection Profile at the Basic Level of Robustness or Greater.

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

6.3 Security Assurance Requirements for the TOE

The Security Assurance Requirements for the TOE are the assurance components of Evaluation Assurance Level 2 (EAL2) augmented by ALC_FLR.2 taken from Part 3 of the Common Criteria. None of the assurance components are refined. The assurance components are listed in Table 6-8.

Table 6-8: Assurance Components

Item	Component	Component Title
1	ADV_ARC.1	Security architecture description
2	ADV_FSP.2	Security-enforcing functional specification
3	ADV_TDS.1	Basic design
4	AGD_OPE.1	Operational user guidance
5	AGD_PRE.1	Preparative procedures
6	ALC_CMC.2	Use of a CM system
7	ALC_CMS.2	Parts of the TOE CM coverage
8	ALC_DEL.1	Delivery procedures
9	ALC_FLR.2	Flaw reporting procedures
10	ASE_CCL.1	Conformance claims
11	ASE_ECD.1	Extended components definition
12	ASE_INT.1	ST introduction
13	ASE_OBJ.2	Security objectives
14	ASE_REQ.2	Derived security requirements
15	ASE_SPD.1	Security problem definition
16	ASE_TSS.1	TOE summary specification
17	ATE_COV.1	Evidence of coverage
18	ATE_FUN.1	Functional testing
19	ATE_IND.2	Independent testing – sample
20	AVA_VAN.2	Vulnerability analysis

Further information on these assurance components can be found in the Common Criteria for Information Technology Security Evaluation (CCITSE) Part 3.

6.4 Security Requirements Rationale

6.4.1 Dependencies Satisfied

Table 6-9 shows the dependencies between the functional requirements including the extended components defined in Section 5. Dependencies that are satisfied by a hierarchical component are denoted by an (H) following the dependency reference.

Table 6-9: TOE SFR Dependencies Satisfied

Item	SFR Short Name	Dependency	Item Ref	Notes
	TOE			
1	FAU_GEN.1-NIAP-0410	FPT_STM.1	27	Satisfied by FIT_PPC_(EXT).1
2	FAU_GEN_(EXT).2	FAU_GEN.1-NIAP-0410	1	
		FIA_UID.1	11, 27	Satisfied by FIA_UID_(EXT).1 and FIT_PPC_(EXT).1
3	FAU_SEL.1-NIAP-0407	FAU_GEN.1-NIAP-0410	1	
		FMT_MTD.1	15	
4	FDP_ACC.1	FDP_ACF.1-NIAP-0407	5	
5	FDP_ACF.1-NIAP-0407	FDP_ACC.1	4	
		FMT_MSA.3	14	Satisfied by FMT_MSA_(EXT).3
6	FDP_RIP.1	None	NA	
7	FIA_ATD.1	None	NA	
8	FIA_UAU_(EXT).1	FIA_UID_(EXT).2	11	
9	FIA_UAU.5	None	NA	
10	FIA_UAU_(EXT).2	None	NA	
11	FIA_UID_(EXT).2	None	NA	
12	FMT_MOF.1	FMT_SMF.1	18	
		FMT_SMR.1	19	
13	FMT_MSA.1	FDP_ACC.1	4	
		FMT_SMF.1	18	
		FMT_SMR.1	19	
14	FMT_MSA_(EXT).3	FMT_MSA.1	13	
		FMT_SMR.1	19	
15	FMT_MTD.1	FMT_SMF.1	18	
		FMT_SMR.1	19	
16	FMT_REV.1(2)	FMT_SMR.1	19	
17	FMT_REV.1(2)	FMT_SMR.1	19	
18	FMT_SMF.1	None	NA	
19	FMT_SMR.1	FIA_UID.1	11, 27	Satisfied by FIA_UID_EXT).1 and FIT_PPC_(EXT).1
20	FPT_OVR_(EXT).1	FPT_TRC_(EXT).1	22	
21	FPT_SIP_(EXT).1	None	NA	
22	FPT_TRC_(EXT).1	FPT_ITT.1	27	This dependency is satisfied by the IT environment due to the lack of cryptography in the TOE and because the DBMS is a software-only TOE.

Item	SFR Short Name	Dependency	Item Ref	Notes
23	FTA_MCS.1	FIA_UID.1	11, 27	Satisfied by FIA_UID_EXT).1 and FIT_PPC_(EXT).
24	FTA_TAH_(EXT).1	None	NA	
25	FTA_TSE.1	None	NA	
26	FTP_ITC_(EXT).1	None	NA	
	IT Environment			
27	FIT_PPC_(EXT).1	None	NA	

Table 6-10 shows the dependencies between the assurance requirements. Dependencies that are satisfied by a hierarchical component are denoted by an (H) following the dependency reference

Table 6-10: TOE SAR Dependencies Satisfied

Item	SAR Short Name	Dependencies	Item Ref
1	ADV_ARC.1	ADV_FSP.1	2 (H)
		ADV_TDS.1	3
2	ADV_FSP.2	ADV_TDS.1	3
3	ADV_TDS.1	ADV_FSP.2	2
4	AGD_OPE.1	ADV_FSP.1	2 (H)
5	AGD_PRE.1	None	N/A
6	ALC_CMC.2	ALC_CMS.1	7
7	ALC_CMS.2	None	N/A
8	ALC_DEL.1	None	N/A
9	ALC_FLR.2	None	N/A
10	ATE_COV.1	ADV_FSP.2	2
		ATE_FUN.1	11
11	ATE_FUN.1	ATE_COV.1	10
12	ATE_IND.2	ADV_FSP.2	2
		AGD_OPE.1	4
		AGD_PRE.1	5
		ATE_COV.1	10
		ATE_FUN.1	11
13	AVA_VAN.2	ADV_ARC.1	1
		ADV_FSP.1	2 (H)
		ADV_TDS.1	3
		AGD_OPE.1	4
		AGD_PRE.1	5

6.4.2 Security Requirements Traced to Objectives

Table 6-11 traces each security requirement back to the security objectives for the TOE.

Table 6-11: Security Assurance Requirements Traced to Objectives

Assurance Component	Objective ID
ADV_ARC.1	O.INTERNAL_ TOE_DOMAINS
ADV_ARC.1	O.PARTIAL_SELF_PROTECTION
ADV_FSP.2	O.DOCUMENTED_DESIGN
ADV_TDS.1	O.DOCUMENTED_DESIGN
AGD_OPE.1	O.ADMIN_GUIDANCE

Assurance Component	Objective ID
AGD_OPE.1	O.ADMIN_GUIDANCE
AGD_PRE.1	O.ADMIN_GUIDANCE
AGD_PRE.1	O.ADMIN_GUIDANCE
ALC_CMS.2	O.CONFIGURATION_IDENTIFICATION
ALC_DEL.1	O.ADMIN_GUIDANCE
ALC_FLR.2	O.CONFIGURATION_IDENTIFICATION
ATE_COV.1	O.PARTIAL_ FUNCTIONAL_TEST
ATE_FUN.1	O.PARTIAL_ FUNCTIONAL_TEST
ATE_IND.2	O.PARTIAL_ FUNCTIONAL_TEST
AVA_VAN.2	O.VULNERABILITY_ANALYSIS

Table 6-12: Security Functional Requirements Traced to Objectives

#	Functional Component	Objective ID
1	FAU_GEN.1-NIAP-0410	O.AUDIT_ GENERATION
2	FAU_GEN_ (EXT).2	O.AUDIT_ GENERATION
3	FAU_SEL.1-NIAP-0407	O.AUDIT_ GENERATION
4	FDP_ACC.1	O.MEDIATE
5	FDP_ACF.1-NIAP-0407	O.MEDIATE
6	FDP_RIP.1	O.RESIDUAL_INFORMATION
7	FIA_ATD.1	O.AUTH
		O.MEDIATE
		O.TOE_ACCESS
8	FIA_UAU_(EXT).2	O.AUTH
9	FIA_UAU.5	O.AUTH
10	FIA_UAU_(EXT).5	O.AUTH
11	FIA_UID_(EXT).2	O.AUTH
12	FMT_MOF.1	O.MANAGE
13	FMT_MSA.1	O.MANAGE
14	FMT_MSA_(EXT).3	O.MANAGE
15	FMT_MTD.1	O.MANAGE
16	FMT_REV.1(1)	O.MANAGE
17	FMT_REV.1(2)	O.MANAGE
18	FMT_SMF.1	O.MANAGE
19	FMT_SMR.1	O.ADMIN_ ROLE
		O.MANAGE
20	FPT_OVR_(EXT).1	O.AVAIL
21	FPT_SIP_(EXT).1	O.PARTIAL_SELF_PROTECTION
22	FPT_TRC_(EXT).1	O.MEDIATE
23	FTA_MCS.1	O.TOE_ACCESS
24	FTA_TAH_ (EXT).1	O.ACCESS_ HISTORY
25	FTA_TSE.1	O.TOE_ACCESS
26	FTP_ITC_(EXT).1	O.PROTCOMM
		OE.PROTCOMM
	IT Environment	
27	FIT_PPC_(EXT).1	OE.OS_PP_VALIDATED OE.AUTH

Table 6-13 demonstrates that the SFRs meet all security objectives for the TOE**.** Rationale for each objective is included in the table.

Table 6-13: All TOE Objectives Met by Security Functional Requirements

Objective ID	SFR ID/Title	Rationale
O.ACCESS_ HISTORY The TOE will store and retrieve information (to authorized users) related to previous attempts to establish a session.	FTA_TAH_ (EXT).1	The TOE must be able to store and retrieve information about previous unauthorized login attempts and the number times the login was attempted every time the user logs into their account. The TOE must also store the last successful authorized login. This information will include the date, time, method, and location of the attempts. When appropriately displayed, this will allow the user to detect if another user is attempting to access their account. These records should not be deleted until after the user has been notified of their access history. (FTA_TAH_(EXT).1)
O.ADMIN_ GUIDANCE The TOE will provide administrators with the necessary information for secure management.	ALC_DEL.1	ALC_DEL.1 ensures that the administrator is provided documentation that instructs them how to ensure the delivery of the TOE, in whole or in parts, has not been tampered with or corrupted during delivery. This requirement ensures the administrator has the ability to begin their TOE installation with a clean (e.g., malicious code has not been inserted once it has left the developer's control) version of the TOE, which is necessary for secure management of the TOE.
	AGD_PRE.1	AGD_PRE.1 ensures the administrator has the information necessary to install the TOE in the evaluated configuration. Often times a vendor's product contains software that is not part of the TOE and has not been evaluated. The Preparative User Guidance (AGD_PRE) documentation ensures that once the administrator has followed the installation and configuration guidance the result is a TOE in a secure configuration.
	AGD_OPE.1	AGD_OPE.1 mandates the developer provide the administrator with guidance on how to operate the TOE in a secure manner. This includes describing the interfaces the administrator uses in managing the TOE, security parameters that are configurable by the administrator, how to configure the TOE's rule set and the implications of any dependencies of individual rules. The documentation also provides a description of how to setup and review the auditing features of the TOE. The guidance must show the administrator how to use the functionality available, review the results of any tests and/or alerts, and act accordingly.
	AGD_OPE.1	AGD_OPE.1 is also intended for non-administrative users, but it could be used to provide guidance on security that is common to both administrators and non-administrators (e.g., password management guidelines).

Objective ID	SFR ID/Title	Rationale
	AGD_OPE.1 AGD_PRE.1	AGD_OPE.1 and AGD_PRE.1 analysis during evaluation will ensure that the guidance documentation is complete and consistent, and notes all requirements for external security measures.
O.ADMIN_ ROLE The TOE will provide authorized administrator roles to isolate administrative actions.	FMT_SMR.1	The TOE will establish, at least, an authorized administrator role. The ST writer may choose to specify more roles. The authorized administrator will be given privileges to perform certain tasks that other users will not be able to perform. These privileges include, but are not limited to, access to audit information and security functions. (FMT_SMR.1)
O.AUDIT_ GENERATION The TOE will provide the capability to detect and create records of security relevant events associated with users.	FAU_GEN.1-NIAP-0410	FAU_GEN.1-NIAP-0410 defines the set of events that the TOE must be capable of recording. This requirement ensures that the administrator has the ability to audit any security relevant events that takes place in the TOE. This requirement also defines the information that must be contained in the audit record for each auditable event. This requirement also places a requirement on the level of detail that is recorded on any additional security functional requirements an ST author adds to this PP.
	FAU_GEN_ (EXT).2	FAU_GEN_(EXT).2 ensures that the audit records associate a user and/or group identity with the auditable event. In the case of authorized users, the association is accomplished with the user ID. In the case of authorized groups, the association is accomplished with the group ID.
	FAU_SEL.1-NIAP-0407	FAU_SEL.1-NIAP-0407 allows the administrator to configure which auditable events will be recorded in the audit trail. This provides the administrator with the flexibility in recording only those events that are deemed necessary by site policy, thus reducing the amount of resources consumed by the audit mechanism.
O.CONFIGURATION_IDENTIFICATION The configuration of the TOE is fully identified in a manner that will allow implementation errors to be identified and corrected with the TOE being redistributed promptly.	ALC_CMS.2	ALC_CMS.2 addresses this objective by requiring that there be a unique reference for the TOE, and that the TOE is labeled with that reference. It also requires that there be a CM system in place, and that the configuration items that comprise the TOE are uniquely identified. This provides a clear identification of the composition of the TOE.
	ALC_FLR.2	ALC_FLR.2 addresses this objective by requiring that there be a mechanism in place for identifying flaws subsequent to fielding, and for distributing those flaws to entities operating the system.
O.DOCUMENTED_DESIGN The design of the TOE is	ADV_FSP.2	ADV_FSP.2 requires that the interfaces to the TOE be documented and specified.

Objective ID	SFR ID/Title	Rationale
adequately and accurately documented	ADV_TDS.1	ADV_TDS.1 requires the high-level design of the TOE be documented and specified and that said design be shown to correspond to the interfaces. ADV_TDS.1 also requires that there be a correspondence between adjacent layers of the design decomposition.
O.INTERNAL_ TOE_DOMAINS The TSF will maintain internal domains for separation of data and queries belonging to concurrent users.	ADV_ARC.1	ADV_ARC.1 provides the security architecture description of the security domains maintained by the TSF that are consistent with the SFRs. Since self-protection is a property of the TSF that is achieved through the design of the TOE and TSF, and enforced by the correct implementation of that design, self-protection will be achieved by that design and implementation.
O.MANAGE The TOE will provide all the functions and facilities necessary to support the authorized administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.	FMT_MOF.1	FMT_MOF.1 requires that the ability to use particular TOE capabilities be restricted to the administrator.
	FMT_MSA.1	FMT_MSA.1 requires that the ability to perform operations on security attributes be restricted to particular roles.
	FMT_MSA_ (EXT).3	FMT_MSA_(EXT).3 requires that default values used for security attributes are restrictive.
	FMT_MTD.1	FMT_MTD.1 requires that the ability to manipulate TOE content is restricted to administrators.
	FMT_REV.1(1) FMT_REV.1(2)	FMT_REV.1 restricts the ability to revoke attributes to the administrator.
	FMT_SMF.1	FMT_SMF.1 identifies the management functions that are available to the authorized administrator.
	FMT_SMR.1	FMT_SMR.1 defines the specific security roles to be supported.
O.MEDIATE The TOE must protect user data in accordance with its security policy.	FDP_ACC.1	The FDP requirements were chosen to define the policies, the subjects, objects, and operations for how and when mediation takes place in the TOE. FDP_ACC.1 defines the Access Control policy that will be enforced on a list of subjects acting on the behalf of users attempting to gain access to a list of named objects. All the operation between subject and object covered are defined by the TOE's policy.
	FDP_ACF.1-NIAP-0407	FDP_ACF.1-NIAP-0407 defines the security attribute used to provide access control to objects based on the TOE's access control policy.
	FIA_ATD.1	FIA_ATD.1 specifies the subject attributes that are used in making access control decisions
	FPT_TRC_ (EXT).1	Replicated TSF data that specifies attributes for access control must be consistent across distributed components of the TOE. The requirement is to maintain consistency of replicated TSF data.
O.PARTIAL_ FUNCTIONAL_TEST The TOE will undergo some security functional testing	ATE_COV.1	ATE_COV.1 requires that there be a correspondence between the tests in the test documentation and the TSF as described in the functional specification.

Objective ID	SFR ID/Title	Rationale
that demonstrates the TSF satisfies some of its security functional requirements.	ATE_FUN.1	ATE_FUN.1 requires that the developer provide test documentation for the TOE, including test plans, test procedure descriptions, expected test results, and actual test results. These need to identify the functions tested, the tests performed, and test scenarios. There require that the developer run those tests, and show that the expected results were achieved.
	ATE_IND.2	ATE_IND.2 requires that the evaluators test a subset of the TSF to confirm correct operation, on an equivalent set of resources to those used by the developer for testing. These sets should include a subset of the developer run tests.
O.PARTIAL_ SELF_PROTECTION The TSF will maintain a domain for its own execution that protects itself and its resources from external interference, tampering, or unauthorized disclosure through its own interfaces.	ADV_ARC.1	ADV_ARC.1 provides the security architecture description of the security domains maintained by the TSF that are consistent with the SFRs. Since self-protection is a property of the TSF that is achieved through the design of the TOE and TSF, and enforced by the correct implementation of that design, self-protection will be achieved by that design and implementation.
	FPT_SIP_(EXT).1	FPT_SIP_(EXT).1 SQL Injection Protection provides protection against the specified common types of SQL Injection attacks.
O.RESIDUAL_INFORMATION The TOE will ensure that any information contained in a protected resource within its Scope of Control is not released when the resource is reallocated.	FDP_RIP.1	FDP_RIP.1 is used to ensure the contents of resources are not available to subjects other than those explicitly granted access to the data.
O.TOE_ACCESS The TOE will provide mechanisms that control a user's logical access to the TOE.	FIA_ATD.1	FIA_ATD.1 defines the attributes of users, including a user ID that is used by the TOE to determine a user's identity and/or group memberships and enforce what type of access the user has to the TOE.
	FTA_MCS.1	FTA_MCS.1 ensures that users may only have a maximum of a specified number of active sessions open at any given time.
	FTA_TSE.1	FTA_TSE.1 allows the TOE to restrict access with finer granularity

Objective ID	SFR ID/Title	Rationale
O.VULNERABILITY_ANALYSIS The TOE will undergo some vulnerability analysis to demonstrate the design and implementation of the TOE does not contain any obvious flaws.	AVA_VAN.2	The AVA_VAN.2 component provides the necessary level of confidence that vulnerabilities do not exist in the TOE that could cause the security policies to be violated. AVA_VAN.2 requires the evaluator to perform a search for potential vulnerabilities in all the TOE deliverables. For those vulnerabilities that are not eliminated by the developer, a rationale must be provided that describes why these vulnerabilities cannot be exploited by a threat agent with a basic attack potential, which is in keeping with the desired assurance level of this TOE. This component provides the confidence that security flaws do not exist in the TOE that could be exploited by a threat agent of basic attack potential to violate the TOE's security policies.
Non-DBMS PP Objectives		
O. AUTH The TOE will provide identification and authentication with the support of the IT environment.	FIA_UAU_ (EXT).2	FIA_UAU_ (EXT).2 specifies that the TOE will require that all users be authenticated before they are allowed to connect to the data.
	FIA_UAU.5	FIA_UAU.5 specifies that the TOE will provide multiple authentication methods.
	FIA_UAU_ (EXT).5	FIA_UAU_ (EXT).5 specifies that the TOE will provide multiple authentication methods with the support of the IT environment.
	FIA_UID_ (EXT).2	FIA_UID_ (EXT).2 specifies that the TOE will require that all users be identified before they are allowed to connect to the data.
	FIA_ATD.1	FIA_ATD.1 specifies the identification and authentication data used in making I&A decisions
O.AVAIL The TOE will provide controlled switchover and a failover capabilities to increase the availability of the database	FPT_OVR_ (EXT).1	FPT_OVR_(EXT).1 specifies that the TOE will provide switchover and failover functionality.
O.PROTCOMM The TOE will protect communication between the DB Server and Postgres Studio and between DB Server and clients in the IT environment using SSL.	FTP_ITC_(EXT).1	FTP_ITC_(EXT).1 specifies that the DB Server will provide a trusted channel between itself and Postgres Studio and between itself and clients in the IT environment.

Table 6-14 provides the rationale for the IT Environment Requirements.

Table 6-14 Rationale for IT Environment Requirements

Environmental Objective	Requirements Addressing the Objective	Rationale
OE.NO_EVIL Sites using the TOE shall ensure that authorized administrators are non-hostile, are appropriately trained and follow all	N/A	This objective does not contain any IT security requirements because it is a non-IT related objective. Thus, the CC does

Environmental Objective	Requirements Addressing the Objective	Rationale
administrator guidance.		not mandate it map to any requirements.
OE.NO_GENERAL_ PURPOSE There will be no general-purpose computing capabilities (e.g., compilers or user applications) available on DMBS servers, other than those services necessary for the operation, administration and support of the DBMS.	N/A	This objective does not contain any IT security requirements because it is a non-IT related objective. Thus, the CC does not mandate it map to any requirements.
OE.OS_PP_VALIDATED The underlying OS has been validated against an NSA sponsored OS PP of at least Basic Robustness.	FIT_PPC_(EXT).1	FIT_PPC_(EXT).1 states the underlying OS must be validated against an OS PP of at least basic robustness.
OE.PHYSICAL Physical security will be provided within the domain for the value of the IT assets protected by the TOE and the value of the stored, processed, and transmitted information.	N/A	This objective does not contain any IT security requirements because it is a non-IT related objective. Thus, the CC does not mandate it map to any requirements.
OE.AUTH The IT environment will provide support for the authentication mechanisms that can be invoked by the TOE.	FIT_PPC_(EXT).1 and N/A	The TOE may rely upon the underlying OS as well as other authentication servers in the IT environment for support for identification and authentication. Requirements have not been specified for other authentication services provided by the IT environment, since this Security Target is compliant with CC Version 3.1R2.
OE.PROTCOMM	FTP_ITC_(EXT).1	The TOE relies upon SSL running on clients in the IT environment for protection of TSF data in transit to the DB Server.

6.4.3 Assurance Rationale

The protection profile is developed at the basic robustness level. The assurance requirements are those recommended in instruction 4 from the Consistency Instruction Manual for Development of US Government Protection Profiles for Use in Basic Robustness Environments, Version 3.0, dated 1 February 2005.

Flaw Remediation is the only requirement not included in any EAL level because it does not add any assurance to the current system, but to subsequent releases. ALC_FLR.2 is included to instruct the vendors on proper flaw remediation techniques.

7 TOE Summary Specification

7.1 IT Security Functions

Section 7.1 describes the specific Security Functions of the TOE. The following sub-sections describe how the TOE meets each SFR listed in Section 6. Table 7-1 summarizes the functions and maps them to security functional requirements for the TOE.

Table 7-1: Security Functions Mapped to Security Functional Requirements

Security Function	Sub-Function		SFR
Security Audit	AU-1	Audit Data Generation	FAU_GEN.1-NIAP-410 FAU_GEN_(EXT).2
	AU-2	Selective Audit	FAU_SEL.1-NIAP-0407
User Data Protection	DP-1	Subset Access Control	FDP_ACC.1 FDP_ACF.1-NIAP-0407
	DP-2	Residual Information Protection	FDP_RIP.1
Identification & Authentication	IA-1	User Attribute Definition	FIA_ATD.1
	IA-2	User and/or Group Identification and Authentication	FIA_UAU_(EXT).2 FIA_UAU.5 FIA_UAU-(EXT).5 FIA_UID_(EXT).2
Security Management	SM-1	Security Roles	FMT_SMR.1
	SM-2	Management of TSF Functions and Data	FMT_MOF.1 FMT_MSA.1 FMT_MTD.1 FMT_SMF.1
	SM-3	Static Attribute Initialization	FMT_MSA._(EXT).3
	SM-4	Revocation	FMT_REV.1(1) FMT_REV.1(2)
Protection of the TSF	PT-1	Internal TSF Consistency	FPT_TRC_(EXT).1
	PT-2	Controlled Switchover / Failover	FPT_OVR_(EXT).1
	PT-3	SQL Injection Protection	FPT_SIP_(EXT).1
TOE Access	TA-1	Limits on Multiple Concurrent Sessions	FTA_MCS.1
	TA-2	TOE Access History	FTA_TAH_(EXT).1
	TA-3	TOE Session Establishment	FTA_TSE.1
Trusted Path /Channels	TP-1	Trusted Channel	FTP_ITC_(EXT).1

7.1.1 Security Audit Functions

7.1.1.1 AU-1: Audit Data Generation

FAU_GEN.1-NIAP-410
FAU_GEN_(EXT).2

The TOE does not allow for the starting-up and shutting-down of the audit functions while database the system is running. Start-up and shut-down of the audit functions can only happen when database server starts/stops. The only way for a cluster owner (OS administrator) to change the auditing configuration is to follow these steps:

- stop the database server,

- modify the postgresql.conf configuration file (this does not effect the auditing until the server restarts), and
- restart the database server.

One cannot startup auditing function without starting the database server; one cannot shutdown auditing without shutting down the database server. As long as the configuration parameters in the postgresql.conf configuration file are set properly, auditing starts when the server starts and a user cannot shutdown auditing without shutting down the database server.

PPAS records audit information into the db server log, a human readable flat file that is stored on the file system of the DB Server. The location of the log file, the maximum size of the file, frequency of log file rollover and other configuration parameters for logging can only be set at server start or by editing the postgresql.conf configuration file.

Auditable events are listed in Table 6-3: Auditable Events from DBMS PP and Table 6-4: Auditable Events for Additional Requirements.

The information in the audit record is selected by the Cluster owner using the log_line_prefix configuration parameter in the postgresql.conf configuration file. The default is an empty string. Each recognized escape is replaced as outlined in Table 7-2 below. Anything else that looks like an escape is ignored. Other characters are copied straight to the log line. Some escapes are only recognized by session processes, and do not apply to background processes.

Table 7-2: DB Server Log Record Prefix Configuration Options

Escape	Effect	Session only
%u	User Name	Yes
%d	Database Name	Yes
%r	Remote Hostname or IP address, and Remote Port	Yes
%h	Remote Host	Yes
%p	Process ID	No
%t	Timestamp without milliseconds	No
%m	Timestamp with milliseconds	No
%i	Command Tag. This is the command which generated the log line.	Yes
%c	Session ID. A unique identifier for each session. It is 2 4-byte hexadecimal numbers (without leading zeros) separated by a dot. The numbers are the Session Start Time and the Process ID, so this can also be used as a space saving way of printing these items.	Yes
%l	Number of the log line for each process, starting at 1	No
%s	Session Start Timestamp	Yes
%v	Virtual transaction ID	No
%x	Transaction ID (0 if none)	Yes
%q	Does not produce any output, but tells non-session processes to stop at this point in the string. Ignored by session processes.	No

%%	Literal %	No

7.1.1.2 AU-2: Selective Audit

FAU_SEL.1-NIAP-0407

The operating system file (Postgresql.conf) is in human readable format that can be directly reviewed/edited by a Cluster owner using an OS text editor provided with the IT environment. Any modification does not take effect until after the server is rebooted.

Postgresql.conf Configuration File

A Cluster owner can configure which events are recorded in the db server log by setting parameters in the postgresql.conf file that is read at server startup. The values used for filtering the db server log file by sql type are:

- **none**: disables sql statement logging
- **ddl**: logs all data definition commands like CREATE, ALTER, and DROP commands
- **mod**: logs all ddl statements, plus INSERT, UPDATE, DELETE, TRUNCATE, and COPY FROM. PREPARE and EXPLAIN ANALYZE statements are also logged if their contained command is of an appropriate type.
- **all**: logs all sql statements

Filtering on success or failure is done the same way as filtering on event types.

- To only include successful events, do not include error events in the filtering options.
- To only include failed events, include error events only.

The db server log file can be filtered by message severity level as follows:

- **DEBUG [1-5]**: Provides information for use by developers.
- **INFO**: Provides information implicitly requested by the user, e.g., during VACUUM VERBOSE.
- **NOTICE**: Provides information that may be helpful to users, e.g., truncation of long identifiers and the creation of indexes as part of primary keys.
- **WARNING**: Provides warnings to the user, e.g., COMMIT outside a transaction block.
- **ERROR**: Reports an error that caused the current transaction to abort.
- **LOG**: Reports information of interest to administrators, e.g., checkpoint activity.
- **FATAL**: Reports an error that caused the current session to abort.
- **PANIC**: Reports an error that caused all sessions to abort.

7.1.2 User Data Protection Functions

7.1.2.1 DP-1: Discretionary Access Control

FDP_ACC.1
FDP_ACF.1 -NIAP-0407

Advanced Server enforces the Discretionary Access Control policy to protect the user data stored in the Advanced Server database using the security attributes of subjects and objects.

Subject security attributes are listed in Table 6-5: Advanced Server Role Security Attributes.

Object security attributes are discussed in sections 7.1.2.1.1 and 7.1.2.1.2 below.

7.1.2.1.1 Objects, Their Operations, and Their Privileges

The objects, their operations, and their privileges are listed in Table 7-3 below.

Table 7-3: Advanced Server Access Control Policy (Objects and Operations)

Object	Operations/ privileges/	Description	Default Access on Objects	Privilege Delegation
Table	**Select** **--Copy To**	Allows accessors to retrieve rows from a table Select also permits the use of "Copy To" command on tables.	Owner only	WITH GRANT OPTION
	Insert **--Copy From**	Allows accessors to create new rows in a table. Insert also allows use of "Copy From" command on tables		
	Update	Allows accessors to update rows in a table		
	Delete	Allows accessors to delete rows from a table		
	Rule	Allows accessors to create a rule on a table This is deprecated syntax, accepted for compatibility only.		
	References	Allows accessors to create foreign key constraint on table		
	Trigger	Allows accessors to create a trigger on a table		
	Truncate	Allows accessors to TRUNCATE the specified table.		
	All privileges	Allows accessors to be granted all available privileges at once		
Column	**Insert**	Allows accessors to insert rows in a table. If the accessor does not have update permission on the on the table, the accessor must specify the specific columns in which values will be inserted and have been granted insert permission on those columns.	**Owner only**	WITH GRANT OPTION

Object	Operations/ privileges/	Description	Default Access on Objects	Privilege Delegation
	Update	Allows accessors to update rows in a table. If the accessor does not have update permission on the on the table, the accessor must specify the specific columns in which values will be inserted and have been granted insert permission on those columns.	**Owner only**	WITH GRANT OPTION
Database	**Temporary, Temp**	Allow accessors to created temporary tables while using the database	PUBLIC	WITH GRANT OPTION
	Connect	Allows accessor to connect to the database		
	Create	Allows accessors to create new schemas		
	All privileges	Allows accessors to be granted all available privileges at once		
Function	**Execute**	Allows accessors the use of the specified function and the use of any operators that are implemented on top of a specified function.	EXECUTE (granted to PUBLIC) Executes with privileges of Definer	WITH GRANT OPTION
	All privileges	Allows accessors to be granted all available privileges at once		
Procedure	**Execute**	Allows accessors to use the specified procedure and the use of any operators that are implemented on top of a specified function.	EXECUTE granted to PUBLIC Executes with privileges of Definer	WITH GRANT OPTION
	All privileges	Allows accessors to be granted all available privileges at once		
Package	**Execute**	Allows accessors the use of all of the package's public procedures, public functions, public variables, records, cursors and other public objects and object types.	EXECUTE granted to PUBLIC Executes with privileges of Definer	WITH GRANT OPTION
	All privileges	Allows accessors to be granted all available privileges at once		
Tablespace	**Create**	Allows accessors to create tables and indexes within the tablespace, and allows databases to be created that have the tablespace as their default tablespace. See Table 7-6 for breakout of Tablespace privileges for object creation.	Owner only	WITH GRANT OPTION
	All privileges	Allows accessors to be granted all available privileges at once		

Object	Operations/ privileges/	Description	Default Access on Objects	Privilege Delegation
Schema	**Create**	Allows accessor to create new objects within the schema. See Table 7-4, for a breakout of schema privileges required for creation/removal of various objects.	Owner only	WITH GRANT OPTION
	Usage	Allows accessor to access objects contained in the specified schema (assuming that the objects' own privilege requirements are also met). Essentially this allows the grantee to "look up" objects within the schema.		
	All privileges	Allows accessors to be granted all available privileges at once		
Language	**Usage**	Allows accessors to use of the specified language for the creation of functions in that language.	PUBLIC	WITH GRANT OPTION
	All privileges	Allows accessors to be granted all available privileges at once		
View	**Select**	Allows accessors to retrieve rows from a view	Owner only Executes with privileges of view definer.	WITH GRANT OPTION
	Rule	Allows accessors to create a rule on the view. This is deprecated syntax, accepted for compatibility only.		
Index	**Create**	Allows table owner or Database Superuser only to create the index.	Owner Only	None
	Delete	Allows index owner or Database Superuser only to delete the index.		
Sequence	**Select**	Allows accessors to retrieve rows from a sequence	PUBLIC	None
	Usage	Allows accessors to use the currval and nextval functions		
	Update	Allows accessors the use of the nextval and setval functions		
Synonym	**Create**	Allows accessors to create a synonym.	PUBLIC	None
	Delete	Allows accessors to delete a synonym	Owner only	

Table Notes:

1) Database Superuser bypasses DAC checks and can perform all operations.

2) The owner of an object has all privileges by default. The right to drop an object or to alter its definition is not identified by a grantable privilege; it is inherent in the owner, and cannot be granted or revoked. The owner also implicitly has all grant options for the object.

3) WITH GRANT OPTION allows the role or user who was granted an object privilege to GRANT the privilege to other roles and users.

7.1.2.1.2 Schema, Database, and Tablespace Creation Privileges

Advanced Server organizes database information in the following entities: schemas, databases, tables and tablespaces. Access rights and privileges are associated with each of these entities. They are described below.

Schema
A schema is a collection of database objects as well as logical structures of data. Schema objects can be created and manipulated with SQL provided that the user doing so has the required privileges.

Schema privileges for objection creation or removal within the schema are specified in Table 7-4: Schema Privileges for Object Creation/Removal.

Table 7-4: Schema Privileges for Object Creation/Removal

Object	Operation	Schema Privilege Required	
		CREATE	USAGE
Table	CREATE	X	
	DROP		X
	ALTER TABLE ADD CONSTRAINT FOREIGN KEY		X
	ALTER TABLE DROP CONSTRAINT		X
Index	CREATE	X	X
	DROP		X
Sequence	CREATE	X	
	DROP		X
Trigger	CREATE	X	X
	DROP		X
View	CREATE	X	X
	DROP		X
Rule	CREATE		X
	DROP		X

Database
A database contains one or more named schemas, which in turn contain tables. Schemas also contain other kinds of named objects, including views, indexes, sequences, rules and triggers.

Database privileges for creation of the database or objects within the database are specified in Table 7-5: Database Privileges for Object Creation

Table 7-5: Database Privileges for Object Creation

Object	Operation	Required Privilege
TEMP TABLE	CREATE	Must have TEMP privilege on the database (this is given to PUBLIC by default)

Object	Operation	Required Privilege
DATABASE	CREATE DATABASE	CREATDB
Schema objects (ie. tables, views, etc.)	CREATE	Must have CREATE privilege on the schema in which the object resides.
PUBLIC DATABASE LINK	CREATE	Must have CREATE PUBLIC DATABASE LINK privilege and CREATE on the database in which the link resides.

Tablespaces

Tablespaces in Advanced Server allow Database Superuser to define locations in the file system where the files containing database objects are stored. Once created, a tablespace can be referred to by name when creating database objects. By using tablespaces, an Database Superuser can control the disk layout of an Advanced Server installation and manage disk space. Tablespaces also allow an Database Superuser to use knowledge of the usage pattern of database objects to optimize performance. Tables, indexes, and entire databases can be assigned to particular tablespaces.

Tablespace privileges for object creation within the tablespace are specified in Table 7-6: Tablespace Privileges for Object Creation.

Table 7-6: Tablespace Privileges for Object Creation

Object	Operation	Required Privilege
Tablespace	CREATE	Database Superuser
Table	CREATE [in the specified tablespace]	CREATE on the target tablespace
Index	CREATE [in the specified tablespace]	CREATE on the target tablespace
Database	CREATE [in the specified tablespace]	CREATE on the target tablespace

7.1.2.1.3 Discretionary Access Control Algorithm

The following ordered rules determine if a user is permitted to perform a requested operation on an object. The process ceases for the requested operation on the object, when either “access is granted” or “access is denied”.

The subject must be successfully authenticated prior to execution of the rules below.

1) Check if the subject has Database Superuser role attribute. If yes, grant access to the requested object.

2) If the requested operation = object creation,
   a) Grant access if the subject is assigned the privilege corresponding to the requested operation of the target object, as specified in Table 7-3 through Table 7-6
   b) Otherwise access denied.

3) If requested operation is access to an existing object, i.e., an operation as specified in Table 7-3,

   a) Check that subject has USAGE privilege on the TARGET schema.

b) Access is denied in absence of USAGE privilege.

4) If the requested operation is EXECUTE on an existing procedure, function or package, check the SECURITY INVOKER/DEFINER attribute

a) If the SECURITY INVOKER/DEFINER attribute = INVOKER, grant access if the user that called the function is assigned the privilege corresponding to the requested operation on the target object, otherwise access denied.

b) If the SECURITY INVOKER/DEFINER attribute = DEFINER, grant access if the user that owns the function is assigned the privilege corresponding to the requested operation on the target object, otherwise access denied.

5) Grant access if the subject explicitly assigned the privilege corresponding to the requested operation of the target object.

6) Grant access if the subject is a member of any role (held in the “in roles” attribute) that has been granted the requested privilege on the target object, either by:

a) Individual user role having been directly granted membership in the group role plus use of SET ROLE command

OR

b) Subject has the INHERIT attribute privilege and is a member of a role that possesses the requested privilege

NOTE: A role with the INHERIT attribute can automatically use whatever database privileges have been granted to all roles it is directly or indirectly a member of. Without INHERIT, membership in another role only grants the ability to SET ROLE to that other role; the privileges of the other role are only available after having done so.

7) If none of the above rules is satisfied, access is denied

7.1.2.2 DP-2: Residual Information Protection (vacuum)

FDP_RIP.1

Residual information protection is enforced through the implementation of “write before read”. Storage for a row is allocated at the time that is in inserted or updated and the new values are written into the allocated space. Data storage and retrieval relies upon indexes and links and there is no way for users to access unallocated disk space.

In order to recover or reuse disk space occupied by updated or deleted rows, Advanced Server provides the Auto-Vacuum Daemon. An UPDATE or DELETE operation does not immediately remove the old version of a row from a table. This approach is necessary to gain the benefits of concurrency control. A row version must not be deleted, while it still may be needed by another transaction. However, eventually, an outdated or deleted row version is no longer of interest to any transaction. The space it

occupies must be reclaimed for reuse by new rows, to avoid infinite growth of disk space requirements. The Auto-Vacuum Daemon monitors table activity and reclaims space as necessary.

Configuration of the Auto-Vacuum Daemon is controlled by parameters in the postgresql.conf configuration file. In the default configuration, the Auto-Vacuum Daemon is enabled and the related configuration parameters are appropriately set.

7.1.3 Identification & Authentication Functions

7.1.3.1 IA-1: User Attribute Definition

FIA_ATD.1

Advanced Server manages database access permissions using the concept of roles. A role can be thought of as either a database user, or a group of database users, depending on how the role is set up. Therefore in Advanced Server, the concept of roles subsumes the concepts of "users" and "groups". A “user” in the Advanced Server is a role with the LOGIN privilege.

By default an Advanced Server user (role) has the security attributes that are specified in Table 6-5: Advanced Server Role Security Attributes

7.1.3.2 IA-2: Identification and Authentication

FIA_UAU_(EXT).2
FIA_UAU.5
FIA_UAU_(EXT).5
FIA_UID_(EXT).2

Advanced Server does not allow access to the TOE until a user has been identified and successfully authenticated by the authentication method set by the authorized administrator

The TOE offers a number of different client authentication methods. The method used to authenticate a particular client connection can be selected on the basis of the client host address, database, and user.

7.1.3.2.1 Authentication Methods

Advanced Server supports the following authentication methods:

- Password (*password* option)
- MD5 password (*md5* option)
- Pluggable Authentication Modules (*pam* option)
- Lightweight Directory Access Protocol (*ldap* option)
- Kerberos (*krb5* option)
- Generic Security Services API (*gss* option)
- Security Service Provider Interface (*sspi* option)
- SSL Certificates (*cert* option)

Password and MD5 Password authentication is provided wholly within the TOE. PAM, LDAP, Kerberos, SSPI, and GSSAPI authentication is provided with the support of the IT environment. “Trust” and “Ident” authentication methods are prohibited in the evaluated configuration.

Password Authentication

Advanced Server database passwords are separate from operating system user passwords. The password for each database user is stored in the pg_authid system catalog. Passwords can be managed with the SQL commands CREATE USER and ALTER USER. By default, if no password has been set up, the stored password is null and password authentication always fails for that user.

The password can be sent across the connection either MD5-hashed (*MD5* option) or in clear-text (*password* option). Since the CC configuration requires the use of SSL encryption for connections, either options can be implemented.

Note: The MD5 implementation is vendor developed to the RFC 1321 specification and is strictly used for password hashing. The MD5 cryptographic function implementation, or module, has not been FIPS certified. The correctness of the cryptographic module used by the TOE is by Vendor assertion; the correctness and conformance of this cryptographic module to the RFC 1321 standard is not be part of this evaluation.

The TOE does not enforce a minimum length password or password strength mechanism. Therefore, the password security policy must be administratively enforced. The following is a minimum guideline for administrators.

Administrators must use passwords that conform to the following policy:

- Minimum length of 8 characters
- At least on character from 3 out of the four character sets:
    - lower case alphabetic character
    - upper case alphabetic character
    - numeric character
    - special character (e.g. '#: or '*')

Administrators and Users must be instructed to keep their passwords secret. They must also be warned to make sure no one observes them entering their password ("Shoulder Surfing").

Some organizations have more stringent identification and authentication requirements, such as longer password length or one character from each of the four character sets. The application administrator should implement the local organization policies.

PAM Authentication

This authentication method operates similarly to password except that it uses a Pluggable Authentication Module (PAM) as the authentication mechanism. The default PAM service name is postgresql and is dependent upon the PAM library supplied with the operating system. PAM is used only to validate user name/password pairs. Therefore the user must already exist in the database before PAM can be used for authentication. A consumer could optionally supply their own service name after the PAM key word in the file pg_hba.conf, but this functionality is outside the scope of the evaluation.

LDAP Authentication

LDAP is a network protocol that provides access to a directory server that serves up authentication information. PPAS supports LDAP as defined in RFC 4510. . LDAP can only be used to validate the user name/password pairs. Therefore the user must already exist in the database before LDAP can be used for authentication. The server and parameters used are specified after the ldap key word in the file pg_hba.conf.

Kerberos Authentication

Kerberos is an industry-standard secure authentication system suitable for distributed computing over a public network. Native Kerberos authentication has been deprecated in Advanced Server and should be used only for backward compatibility. Native Kerberos authentication is outside the scope of the evaluation. The evaluated TOE can direct Kerberos requests to a Kerberos server in the IT environment. While PostgreSQL supports both Kerberos 4 and Kerberos 5, only Kerberos 5 is recommended. Kerberos 4 is considered insecure and no longer recommended for general use.

GSSAPI Authentication

GSSAPI is an industry-standard protocol for secure authentication defined in RFC 2743. PPAS supports GSSAPI with Kerberos authentication according to RFC 1964. GSSAPI provides automatic authentication (single sign-on) for systems that support it.

SSPI Authentication

SSPI is a Windows technology for secure authentication with single sign-on. Advanced Server uses SSPI in negotiate mode, which uses Kerberos when possible and automatically falls back to NTLM in other cases. SSPI authentication only works when both server and client are running Windows.

SSL Cert Authentication

The cert authentication method uses SSL client certificates to perform authentication. It is therefore only available for SSL connections. When using the cert authentication method, the server will require that the client provide a valid certificate. No password prompt will be sent to the client. The cn attribute of the certificate will be compared to the requested database username, and if they match the login will be allowed. Username mapping can be used to allow cn to be different from the database username.

7.1.3.2.2 pg_hba.conf Configuration File

Client authentication is controlled by a configuration file, which is traditionally named pg_hba.conf and stored in the database cluster's data directory. A default pg_hba.conf file is installed when the data directory is initialized.

The general format of the pg_hba.conf file is a set of records, one per line. Each record specifies a connection type, a client IP address range (if relevant for the connection type), a database name, a user name, and the authentication method to be used for connections matching these parameters. The first record with a matching connection type, client address, requested database, and user name is used to perform authentication.

The pg_hba.conf records are examined sequentially for each connection attempt, so the order of the records is significant. Earlier records should have tighter connection match parameters and weaker authentication methods, while later records should have looser match parameters and stronger authentication methods.

The pg_hba.conf file is read on start-up and when the main server process receives a SIGHUP signal. If the pg_hba.conf file is edited on an active system, it is necessary to signal the server (using pg_ctl reload or kill -HUP) to make it re-read the configuration file.

To connect to a particular database, a user must not only pass the pg_hba.conf checks, but must have the CONNECT privilege for the database. In order to restrict which users can connect to which databases, it is usually easier to control this by granting/revoking CONNECT privilege than by putting the rules into pg_hba.conf file entries.

A pg_hba.conf file record can have one of the seven formats:

- local database user auth-method [auth-options]
- host database user CIDR-address auth-method [auth-options]
- hostssl database user CIDR-address auth-method [auth-options]
- hostnossl database user CIDR-address auth-method [auth-options]
- host database user IP-address IP-mask auth-method [auth-options]
- hostssl database user IP-address IP-mask auth-method [auth-options]
- hostnossl database user IP-address IP-mask auth-method [auth-options]

The meaning of the fields is provided in Table 7-7 below.

Table 7-7: pg_hba.conf Configuration File

Field	Meaning
local	Matches connection attempts using Unix-domain sockets. Without a record of this type, Unix-domain socket connections are disallowed.
host	Matches connection attempts made using TCP/IP. Host records match either SSL or non-SSL connection attempts.
hostssl.	This record matches connection attempts made using TCP/IP, but only when the connection is made with SSL encryption.
hostnossl	Matches connection attempts made over TCP/IP that do not use SSL.
database	Specifies which database names this record matches.
user	Specifies which database user names this record matches.
CIDR-address	Specifies the client machine IP address range that this record matches.
auth-method	Specifies the authentication method to use when connecting via this record.
	auth-method Values
md5	Require the client to supply an MD5-encrypted password for authentication.
password	Require the client to supply an unencrypted password for authentication. Not in the evaluated configuration
pam	Authenticate using the Pluggable Authentication Modules (PAM) service
ldap	Authenticate using LDAP to a central server.
cert	Authenticate using SSL client certificates
krb5	Use Kerberos V5 to authenticate the user. Only available for TCP/IP connections.
gss	Use GSSAPI to authenticate the user. Only available for TCP/IP connections.
sspi	Use SSPI to authenticate the user. Only available on Windows.
ident	Obtain the operating system user name of the and check if the user is allowed to connect as the requested database user using the ident mapping table. Not to be used in the evaluated

Field	Meaning
	configuration.
trust	This method allows anyone that can connect to the database server to login as any PostgreSQL user they like, without the need for a password. Not to be used in the evaluated configuration.
reject	Reject the connection unconditionally or conditionally. This is useful for filtering out certain hosts. This is not considered an authentication mechanism.
auth-options	The meaning of this optional field depends on the chosen authentication method

7.1.4 Security Management Functions

7.1.4.1 SM-1 Security roles

FMT_SMR.1

Advanced Server maintains the following roles:

- Authorized administrator; and
- Customized roles created by the authorized administrator based on security attributes (SUPERUSER, CREATEROLE, AND CREATEDB privileges) defined in FIA_ATD.1.

The “authorized administrator” role in the DBMS PP is called the “Database Superuser” in Advanced Server.

The only pre-defined role is the Database Superuser. The Database Superuser and other roles with the CREATEROLE privilege can create other roles in Advanced Server,

The security attributes associated with Advanced Server Roles are specified in Table 6-5: Advanced Server Role Security Attributes

The following security attributes make a role trusted: SUPERUSER, CREATEROLE, and CREATEDB.

The Advanced Server concept of a role encompasses that of user (a role with the LOGIN privilege), group (a role that is a container for other roles) and the traditional concept of a role (a set of privileges to operate on objects that are associated with a user). A group is a role that has members.

In addition to the database roles maintained by Advanced Server, the TOE relies on the operating system for the role of System administrator. A Cluster owner is a trusted OS user who is authenticated to the operating system of the DBServer and who has permission to interact with that OS to perform operations necessary for the installation, configuration and maintenance of the TOE, such as the modification of configuration files.

The Database Superuser or a user with the CREATEROLE privilege can create new roles using the CREATE ROLE command. These roles can either be additional trusted roles as described above, new LOGIN roles, or group roles that do not have LOGIN privileges. Group roles are created for the purpose of grouping and managing sets of object privileges.

To use a group role to manage object privileges:

1. A superuser (or user with the CREATEROLE security attribute) creates a group role.

2. Object owners grant object privileges to the new group role, allowing all members of the group role to access the object.

For example, an administrator may create a group named 'clerks' and assign SELECT privileges to that group on a table that contains customer information. The administrator may opt to deny the group role 'clerks' access to sensitive information such as social security number.

7.1.4.2 SM-2: Management of TSF Data and Functions

FMT_MOF.1
FMT_MSA.1
FMT_MTD.1
FMT_SMF.1

The management functions provided by the TOE are listed in the following tables:

- Table 6-6: Management of Security Functions Behavior and
- Table 6-7: Management of TSF Data

The management of security attributes for discretionary access control is discussed in Section DP-1: Discretionary Access Control

There are several administrative interfaces for managing the TOE. The Database Superuser and other Advanced Server trusted roles use the TOE's own interfaces (the Developer Studio GUI, the DBA Management Server, and the command line interface to perform management functions.

Database Superusers and the database owner can view the pg_settings catalog table to view the name, value and description of almost all database runtime configuration parameters. All these parameters can be changed by editing the postgresql.conf file and some parameters can also be changed at runtime by issuing an ALTER DATABASE command. Run-time parameters changed via the ALTER DATABASE command are only implemented on new sessions created for that particular database, not for any current session.

Some DBMS configuration options are set by the "Cluster owner", the trusted operating system user created during installation and has been granted the necessary permissions to modify Advanced Server configuration files. The Cluster owner can also start, stop, and restart the server.

Changes to the postgresql.conf file made at the command line by the Cluster owner become persistent and become active only after the server is restarted.

The Cluster owner, OS user with access to Advanced Server configuration files, can accomplish the following tasks:

#	Operation	Function	OS User
1	Determine the behavior of	Specification of Auditable DBMS events	Cluster owner
2	Determine the behaviour of	Residual information protection function	Cluster owner
3	Determine the behaviour of	Authentication functions	Cluster owner
4	Determine the behaviour of	User identification function	Cluster owner

#	Operation	Function	OS User
5	Determine the behaviour of	Replication	Cluster owner
6	Determine the behaviour of	Controlled switchover	Cluster owner
7	Determine the behaviour of	Failover	Cluster owner
8	Determine the behaviour of	Session establishment	Cluster owner
9	Determine the behaviour of	Intrusion detection	Cluster owner
10	View	audit records stored at OS level	Cluster owner
11	Set (modify)	Startup configuration parameters stored in OS files	Cluster owner
12	Set (modify)	Maximum allowed number of concurrent user sessions	Cluster owner
13	Set (modify)	Run-time configuration parameters (persistent beyond server restart)	Cluster owner
14	Set (modify)	Authentication data	Cluster owner
15	Reload	Database server configuration parameters	Cluster owner
16	Start, Stop and Restart	Database server	Cluster owner

7.1.4.3 SM-3: Static Attribute Initialization

FMT_MSA_(EXT).3

The TSF provides restrictive default values for the security attributes that are used to enforce the Discretionary Access Control policy.

Upon creation of a database object that can contain user data, only the object's owner has access to it. The owner must explicitly grant access to other roles. This includes databases, tables, tablespaces, and schema. Some types of objects such as synonym language have public access when they are created. However, these database objects are not used for storing user data. For functions, procedures, and packages will only execute successfully, if the user has already been granted access to the underlying tables where data is stored. Please see Table 7-3: Advanced Server Access Control Policy (Objects and Operations) for details.

7.1.4.4 SM-4: Revocation

FMT_REV.1(1)
FMT_REV.1(2)

The GRANT and REVOKE SQL commands are used to grant and revoke security attributes to users. Only the Database Superuser or a role with the CREATEROLE privilege can execute these commands. The command takes effect the next time that the user logs in.

Object security attributes or privileges are controlled by the owner of the object or a role to whom the owner has granted the privilege with the grant option. Object privileges are revoked using the REVOKE command. Users can only revoke privileges that they granted, either directly or indirectly. I.e., if a user revokes an access privilege from a role, but the role was granted the same privilege by another user, the role still has the privilege.

Privileges can be granted indirectly. This happens when a user who was granted a privilege with the GRANT OPTION grants that privilege to a third user. Privileges that are granted by another user using the GRANT OPTION are called dependent privileges. If an object privilege is revoked using the CASCADE option, dependent privileges are also revoked.

When object privileges are revoked, the revocation takes effect the next time the role opens the object. Queries already in progress may be long running and will continue until they are completed unless some other action is taken.

7.1.5 Protection of the TSF Functions

7.1.5.1 PT-1: Internal TSF Consistency

FPT_TRC_(EXT).1

The Advanced Server TOE includes the Slony-I open source asynchronous replication system for the replication of Advanced Server databases. Slony-I replication is a trigger-based replication solution. It provides a “one master to many subscribers” replication system with cascading replication support.

Slony-I provides database replication services between nodes in a cluster. It is able to replicate large databases to a limited number (on the order of a dozen) of subscriber systems.

Slony-I uses the following main abstractions:

Cluster: Named set of Advanced Server database instances; replication takes place between those databases.

Node: Named Advanced Server database that will be participating in replication.

Replication Set: Set of tables and sequences that are to be replicated between nodes in a Slony-I cluster. There may b several sets, and the "flow" of replication does not need to be identical between those sets.

Master Node or Master Provider: Only place where user applications are permitted to modify data in the tables that are being replicated. There is one origin node per replication set.

Origin node: Another term for the Master Node.

Subscribers: Other nodes in the cluster that subscribe to the replication set, indicating that they want to receive the data. (The origin node is never a "subscriber.") However, Slony-I supports the notion of cascaded subscriptions, that is, a node that is subscribed to some set may also behave as a "provider" to other nodes in the cluster for that replication set.

slon Daemon: Process to manage replication activity for a node, There is one slon() daemon process per node.

slonik Configuration Processor: Processes scripts that are used to submit events to update the configuration of a Slony-I cluster. This includes such things as adding and removing nodes, modifying communications paths, and adding or removing subscriptions. Actions are performed using SLONIK commands such as SLONIK DROP NODE, SLONIK FAILOVER, and SLONIK MOVE SET

The slon and the slonik instances need no special connections or protocols to communicate with one another; they merely need access to the Advanced Server databases, connecting as a "Database Superuser" that has the ability to update system tables.

Slony-I implements asynchronous replication, using triggers to collect table updates, where a single origin may be replicated to multiple subscribers including cascaded subscribers.

On the "origin" node for each replicated table, an additional trigger is added which runs the stored procedure schemadoclogtrigger( ). On each subscriber node, tables are augmented with a trigger that runs the schemadocdenyaccess( ) function; this function prevents anything other than the slon process from updating data in replicated tables. In addition, any other triggers and rules on replicated tables are suppressed on the subscribers: This is done by pointing them, in the system table, to the primary key index instead of to the table itself.

Hosts that are to replicate between one another must have bidirectional network communications between the two database servers.

All the servers used within the replication cluster need to have their Real Time Clocks in sync. The best practice is to run using TZ=UTC or TZ=GMT.

SLONY-I has run-time configuration parameters to control the timing of replication and to ensure that replication occurs in a timely manner. These parameters can be set either on the command line or in a configuration file. Table 7-8 below provides a name, description, range of values, and default value for the replication timing-related SLONY-I configuration parameters.

Table 7-8 SLONY-I Replication Timing Parameters

Parameter	Description	Range	Default
sync_interval	Check for updates at least this often in milliseconds.	10 to 60000	100
sync_interval_timeout	Maximum amount of time in milliseconds before issuing a SYNC event,	0 to 120000	1000
sync_group_maxsize	Maximum number of SYNC events that a subscriber node will group together if a subscriber falls behind	0 to 10000	20
desired_sync_time	Maximum time planned for grouped SYNCs. If replication is behind, slon will try to increase numbers of syncs done targeting that they should take this quantity of time to process. If the value is set to 0, this logic will be ignored.	10000 to 600000	60000
lag_interval	Indicates an interval by which this node is to lag its providers. If set, events are ignored until they reach the age of this interval. If the value is left empty, this logic is ignored.	0	0
remote_listen_timeout	How long, in milliseconds, should the remote listener wait before treating the event selection criteria as having timed out?	30 to 30000	300

sync_interval and sync_interval_timeout work together. On an origin node, sync_interval is the minimum time period that will be covered by a SYNC, and during periods of heavy application activity, it may be that a SYNC is being generated every sync_interval milliseconds. On that same origin node, there may be quiet intervals, when no replicatable changes are being submitted. A SYNC will be induced, anyways, every sync_interval_timeout milliseconds.

The developer warns against setting the lag_interval parameter, because when all nodes need to synchronize during a controlled switchover or failover, the other nodes have to wait for the lagging node.

7.1.5.2 PT-2: Controlled Switchover / Failover

FPT_OVR_(EXT).1

Controlled switchover and failover is done in Advanced Server using the open source Slony-I asynchronous replication system described in the previous section.

The subscriber node is a replicated copy of the master node. Replication occurs asynchronously, and is based on the configured triggers of After (UPDATE, INSERT, or DELETE). It is possible that there is a delay between when transactions are committed on the master node and when they are committed on the subscriber node.

In order to perform controlled switchover or failover, the master or origin node must have been replicating data to a subscriber or backup node.

Controlled Switchover

Controlled switchover can be performed when the master or origin node is still operational. Controlled switchover is done using SLONIK MOVE SET command.

Applications that were connected to the old database must drop those connections and establish new connections to the database that has been promoted to the master role. Once all the committed transactions from the old master node have been replicated on the new master node, the server hosting the old master node can be shutdown. During controlled switchover, no transactions are lost.

Failover

If a serious problem occurs on the master or origin node, it may be necessary to failover to a subscriber backup server. Failover is done using the SLONIK FAILOVER command.

Transactions committed on the master or origin node, but not applied to the subscriber node, are lost. This is a highly undesirable circumstance, as these transactions may have been reported as "successful" to applications and users. Failover should be considered only as a last resort.

After the failover is complete and new master node is accepting write operations against tables, all remnants of the old master node's configuration information should be removed using the SLONIK DROP NODE command: (This is to ensure that two nodes do not both respond as the master node. This situation might occur if the old master node became inaccessible due to a network outage as opposed to failure of data storage.)

NOTE: Although PPAS may provide failover/switchover for crossover platforms configuration (Linux ↔ Windows), that configuration is not recommended and it is not supported by EnterpriseDB. Therefore, crossover platform configuration was not included in the CC evaluated configuration and was not tested during this evaluation. The failover/switchover function was only evaluated for the following configurations of the EDB server:

- *Linux RH5 <-> Linux RH5*
  and
- *Windows 2003 <-> Windows 2003.*

7.1.5.3 PT-3: SQL/Protect

FPT_SIP_(EXT) SQL Injection Protection

A "SQL injection attack" is an attempt to compromise a database by running SQL statements whose results provide clues to the attacker as to the content, structure, or security of that database. SQL/Protect provides a layer of security in addition to normal database security policies by examining incoming queries for the following common SQL injection attacks:

- Unauthorized relations,
- Utility commands,
- SQL tautology, and
- Unbounded DML statements

These types of SQL injection attacks are discussed in more detail below.

Modes

SQL/Protect has the following configurable modes:

- **Learn:** SQL/Protect learns the normal behavior for a user or role by recording the relations
- **Passive:** SQL/Protect records and displays potential SQL injection attacks to the Database Superuser, and
- **Active:** SQL/Protect blocks potentially dangerous queries as well as recording them and displaying them to the Database Superuser.

The same mode applies to all monitored roles.

Types of SQL Injection Attacks

Unauthorized Relations: Postgres Plus Advanced Server allows authorized administrators to restrict access to relations (tables, views, etc.), but some authorized administrators do not perform this tedious task. SQL/Protect provides a "learn" mode that tracks the relations a user accesses. This allows authorized administrators to examine the workload of an application, and for SQL/Protect to learn which relations an application should be allowed to access for a given user. When SQL/Protect is switched to either "passive" or "active" mode, the incoming queries are checked against the list of learned relations.

Note: As a reminder the term "authorized administrators" is used when both the "Database Superuser" and the "Cluster owner" are required for implementation. See Section 1.4.4 Users for further detail.

Utility Commands: A common technique used in SQL injection attacks is to run utility commands, which are typically SQL Data Definition Language (DDL) statements. An example is creating a user-defined

function that has the ability to access other system resources. SQL/Protect can block the running of all utility commands, which are not normally needed during standard application processing.

SQL Tautology: A common technique used in SQL injection attacks is issuing a tautological WHERE clause condition (that is, using a condition that is always true) such as “WHERE password = 'x' OR 'x'='x'”. Attackers often start identifying security weaknesses using this technique. SQL/Protect can block queries that use a tautological conditional clause.

Unbounded DML Statements: A dangerous action taken during SQL injection attacks is the running of unbounded DML statements. These are UPDATE and DELETE statements with no WHERE clause. For example, an attacker may update all users' passwords to a known value or initiate a denial of service attack by deleting all of the data in a key table. SQL/Protect can block UPDATE or DELETE statements with no WHERE clause.

Protected Roles

A "protected role" is a Postgres Plus user or group that the Database Superuser has chosen to monitor using SQL/Protect. (In Postgres Plus, users and groups are collectively referred to as "roles".) Each protected role can be customized for the types of SQL injection attacks for which it is to be monitored, thus providing different levels of protection by role.

Attack Attempt Statistics

Each usage of a command by a protected role that is considered an attack by SQL/Protect is recorded in a statistics view that can be monitored to identify the start of a potential attack. Statistics are collected by type of SQL injection attack.

7.1.6 TOE Access Functions

7.1.6.1 TA-1 Limits on multiple concurrent sessions

FTA_MCS.1
The number of multiple concurrent sessions per role is determined by the “connection limit” role security attribute. The “connection limit” is checked during session establishment.

“-1” means that an unlimited number of connections are allowed and is the default value.

The Database Superuser, customized role with the CREATEROLE attribute, and the Cluster owner can change this parameter.

7.1.6.2 TA-2 TOE access history

FTA_TAH_ (EXT).1)

The accessHistory() function uses functionality provided by the database server to provide a way for a user (or a database superuser) to retrieve information about connection history. The history will include all connection and disconnection attempts recorded in the specified log file for the calling role.

accessHistory() is a security-definer function owned by a superuser. When a nonsuperuser calls the function, he assumes sufficient privileges to execute the accessHistory() function only until the function completes.

accessHistory() depends on the database server to manage the privileges that designate which connection history a specific role may view; a non-superuser role can view only his own history.

7.1.6.3 TA-3 TOE Session Establishment

FTA_TSE.1

The TSF can deny session establishment based on user identity, group identity, database name, Host IP address, and/or subnet address contained in the pg_hba.conf, and maximum number of connections allowed on server (postgres.conf) .

During session establishment, the database name, username and source IP combination are validated against the pg_hba.conf file. Once this process succeeds, a connection attempt is made to the database at which time the server determines whether the *maximum number of connections allowed on the server* threshold is met. If the number of session exceeds the threshold limit the connection is refused. If the threshold limit has not been reached the the user (role with LOGIN attribute value) is authenticated,

The pg_hba.conf file and is stored in the database cluster's data directory (HBA stands for host-based authentication) and is only modifiable by the Cluster owner at the OS level. A default pg_hba.conf file is installed when the data directory is initialized.

The general format of the pg_hba.conf file is a set of records, one per line. Blank lines are ignored, as is any text after the # comment character. A record is made up of a number of fields which are separated by spaces and/or tabs. Fields can contain white space if the field value is quoted. Records cannot be continued across lines.

Each record specifies a connection type, a client IP address range (if relevant for the connection type), a database name, a user name, and the authentication method (*password, md5, gss, sspi, krb5, ldap, cert,* and to explicitly deny connection *reject*) to be used for connections matching these parameters. The first record with a matching connection type, client address, requested database, and user name is used to perform authentication or explicitly reject (deny) the connection. There is no "fall-through" or "backup": if one record is chosen and the authentication fails, subsequent records are not considered. If no record matches, access is denied.

An example of where a host is being denied session establishment is as follows:

```
# In the absence of preceding "host" lines, these two lines will
# reject all connections from 192.168.54.1 (since that entry will be
# matched first), but allow Kerberos 5 connections from anywhere else
# on the Internet.  The zero mask means that no bits of the host IP
# address are considered so it matches any host.
#
# TYPE  DATABASE    USER        CIDR-ADDRESS          METHOD
  host        all          all        192.168.54.1/32         reject
  host        all          all           0.0.0.0/0            krb5
```

host: This record matches connection attempts made using TCP/IP. host records match either SSL or non-SSL connection attempts.

Database: Specifies which database name(s) this record matches. The value all specifies that it matches all databases.

User: Specifies which database user name(s) this record matches. The value all specifies that it matches all users.

CIDR-address: Specifies the client machine IP address range that this record matches. This field contains an IP address in standard dotted decimal notation and a CIDR mask length. (IP addresses can only be specified numerically, not as domain or host names.) The mask length indicates the number of high-order bits of the client IP address that must match. Bits to the right of this must be zero in the given IP address. There must not be any white space between the IP address, the /, and the CIDR mask length.

Typical examples of a *CIDR-address* are 172.20.143.89/32 for a single host, or 172.20.143.0/24 for a small network, or 10.6.0.0/16 for a larger one. To specify a single host, use a CIDR mask of 32 for IPv4 or 128 for IPv6. In a network address, do not omit trailing zeroes.

auth-method: Specifies the authentication method to use when a connection matches this record. The possible choices are summarized here.

Parameter	Description
reject	Reject the connection unconditionally. This is useful for "filtering out" certain hosts from a group.
password	Require the client to supply an unencrypted password for authentication. Since the password is sent in clear text over the network, this should not be used on untrusted networks.
md5	Require the client to supply an MD5-encrypted password for authentication.
ldap	Authenticate using an LDAP server.
pam	Authenticate using the Pluggable Authentication Modules (PAM) service provided by the operating system.
gss	Use GSSAPI to authenticate the user. This is only available for TCP/IP connections.
krb5	Use Kerberos V5 to authenticate the user. This is only available for TCP/IP connections.
sspi	Use SSPI to authenticate the user. This is only available on Windows.
cert	Authenticate using SSL client certificates.
	Authentication methods not allowed in CC configuration
trust	Allow the connection unconditionally. This method allows anyone that can connect to the PostgreSQL database server to login as any PostgreSQL user , without the need for a password.
ident	Obtain the operating system user name of the client (for TCP/IP connections by contacting the ident server on the client, for local connections by getting it from the operating system) and check if it matches the requested database user name. (Not allowed in CC configuration)

7.1.7 Trusted Path/Channels

7.1.7.1 TP-1 Partial Trusted Channels

FTP_ITC_(EXT).1

The PPAS Database Server has native support for using SSL connections to encrypt client/server communications. This requires that OpenSSL is installed on both the client and server systems and that support in PostgreSQL is enabled at build time. With SSL support compiled in, the PostgreSQL server can be started the parameter ssl to “on” in the postgresql.conf configuration file. The Server will listen for both standard and SSL connections on the same TCP port and will negotiate with any connecting client on whether to used SSL. By default, this is at the client’s option. However, the server can be set up to require the use of SSL for some or all connections using the pg_hba.conf file.

Although SSL support is compiled into the Database Server and SSL is configured and managed through the TOE, the TOE still relies upon OpenSSL on the underlying platform to perform cryptographic operations.

Postgres Studio can be configured to use SSL to connect to a Database Server using the “New Server Registration” Dialog.

8 Appendix A: Text Omitted from DBMS PP for Refinements

This section contains refinements where text was omitted. Omitted text is shown as bold text within parenthesis. The actual text of the functional requirements as presented in Section 5 has been retained.

FAU_SEL.1.1-NIAP-0407 **Refinement**: The TSF shall **(be able to)** allow only the administrator to include or exclude auditable events from the set of audited events based on the following attributes:

a) user identity and/or group identity,
b) event type,
c) object identity,
d) [selection: “subject identity”, “host identity”, “none”];
e) [success of auditable security events;
f) failure of auditable security events; and
g) [selection: [assignment: list of additional criteria that audit selectivity is based upon], “no additional criteria”].]

FMT_MSA.1.1 **Refinement**: The TSF shall enforce the [Discretionary Access Control policy] to restrict the ability to [manage] all the security attributes to [authorized administrators] **(including top-level objects (e.g., tables) and sub-level objects (e.g., rows, columns, cells)).**

Application Note: The ST author should ensure that all attributes identified in FIA_ATD.1 are adequately managed and protected for top-level objects and sub-level objects.

FTA_TSE.1.1 **Refinement**: The TSF shall be able to deny session establishment based on [attributes that can be set explicitly by authorized administrator(s), including user identity and/or group identity **(, time of day, day of the week)**], and [assignment: list of additional attributes].

9 Appendix B: DBMS PP References

[1] Common Criteria Implementation Board, Common Criteria for Information Technology Security Evaluation, CCIB-98-026, Version 2.1, August 1999 [1a] Common Criteria Implementation Board, Common Criteria for Information Technology Security Evaluation, CCMB-2006-09, Version 3.1, September 2006

[2] Department of Defense Chief Information Officer, Guidance and Policy for Department of Defense Information Assurance Memorandum No. 6-8510 dated 16 June 2000

[2a] Department of Defense Directive 8500.1, "Information Assurance," October 24, 2002

[2b] Department of Defense Instruction 8500.2, "Information Assurance," February 6, 2003

[3] National Security Agency, Protection Profile for Single-level Operating Systems in Environments Requiring Medium Robustness Version 1.22, 23 May 2001

[3a] National Security Agency, Protection Profile for Single-level Operating Systems in Environments Requiring Medium Robustness Version 1.91, 16 March 2007

[4] Department of Defense Standard, Department of Defense Trusted Computer System Evaluation Criteria (Orange Book), December 1985

[5] Trusted Product Evaluation Program (TPEP) Trusted Computer System Evaluation Criteria (TCSEC) Interpretations

[6] National Computer Security Center, Trusted Database Management System Interpretation of the Trusted Computer System Evaluation Criteria, NCSC-TG-021 Version-1, April 1991

[7] Security Agency Information Assurance Solutions Technical Directors, Information Assurance Technical Framework, Release 3.1, September 2002

[8] Protection Profile for Operating Systems Implementing Commercial Security, Version 1.0, dated 27 December 2001

[9] Protection Profile Review Board, Protection Profile Consistency Guidance for Basic Robustness, Version 3.0, dated 1 February 2005