National Information Assurance Partnership

Common Criteria Evaluation and Validation Scheme Validation Report

Postgres Plus Advanced Server v8.4

Report Number: CCEVS-VR-VID10412-2011
Dated: July 29, 2011
Version: 1.0

National Institute of Standards and Technology
Information Technology Laboratory
100 Bureau Drive
Gaithersburg, MD 20899

National Security Agency
Information Assurance Directorate
9800 Savage Road STE 6940
Fort George G. Meade, MD 20755-6940

ACKNOWLEDGEMENTS

Validation Team

Jandria Alexander

Aerospace Corporation

Jim Brosey

Orion Security Solutions

Common Criteria Testing Laboratory

Dragua Zenelaj

CygnaCom Solutions

Much of the material in this report was extracted from evaluation material prepared by the CCTL. The CCTL team deserves credit for their hard work in developing that material. Many of the product descriptions in this report were extracted from the Enterprise DB Postgres Plus Advanced Server v8.4 Security Target.

Table of Contents

List of Figures

1. Executive Summary

This Validation Report (VR) documents the evaluation and validation of the product Postgres Plus Advanced Server v8.4.

Postgres Plus Advanced Server v8.4 (PPAS) is a relational database management system (RDBMS) based on PostgreSQL, an open source database. PPAS provides these security functions: 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 evaluation was performed by the CygnaCom Common Criteria Testing Laboratory (CCTL), and was completed in July 2011. The information in this report is derived from the Evaluation Technical Report (ETR) and associated test reports, all written by the CygnaCom CCTL.

The evaluation team determined that the product is Common Criteria version 3.1 R3 [CC] Part 2 extended and Part 3 conformant, and meets the assurance requirements of EAL 2 augmented by ALC_FLR.2 Flaw reporting procedures from the Common Methodology for Information Technology Security Evaluation, Version 3.1 R3, [CEM]. 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.

The evaluation and validation were consistent with National Information Assurance Partnership (NIAP) Common Criteria Evaluation and Validation Scheme (CCEVS) policies and practices as described on their web site www.niap-ccevs.org. The Security Target (ST) is EnterpriseDB Postgres Plus Advanced Server v8.4 Security Target.

This VR is not an endorsement of the IT product by any agency of the U.S. Government and no warranty of the IT product is either expressed or implied.

2. Identification

Target of Evaluation:	Postgres Plus Advanced Server v8.4
Developer:	EnterpriseDB Corporation
CCTL:	CygnaCom Solutions 7925 Jones Branch Dr, Suite 5400 McLean, VA 22102-3321
Evaluators:	Dragua Zenelaj
Validation Scheme:	National Information Assurance Partnership CCEVS
Validators:	Jandria Alexander Jim Brosey
CC Identification:	Common Criteria for Information Technology Security Evaluation, Version 3.1 R3, July 2009
Interpretations:	There were no applicable interpretations used for this evaluation.
CEM Identification:	Common Methodology for Information Technology Security Evaluation, Version 3.1 R3, July 2009
PP:	US Government Protection Profile for Database Management Systems in Basic Robustness Environments, Version 1.2, July 25, 2007
Evaluation Class:	Evaluation Assurance Level (EAL) 2 augmented with ALC_FLR.2
Completion Date:	July 2011

3. Security Policy

The TOE's security policy is expressed in the security functional requirements identified in Section 6.1 of the ST. Potential users of this product should confirm that functionality implemented is suitable to meet the user's requirements.

The TOE provides the following security features:

3.1. Security Audit Functions

Postgres Plus Advanced Server v8.4 (PPAS) generates audit records for security relevant events. The TOE provides the capability to select auditable events based on settings in a system configuration files.

3.2. User Data Protection Functions

PPAS 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 PPAS, 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.

3.3. Identification and Authentication Functions

PPAS ensures that users are identified and authenticated by a TOE supported 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 Operational Environment. Note that the use of the "Trust" or "Ident" authentication methods is 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.*

3.4.Security Management Functions

PPAS 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 PPAS.

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

3.5.Protection of TOE Security Functions

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.

3.6.TOE Access Functions

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

The OTE 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.

3.7.Partial Trusted Communication Functions

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

4. Threats, OSPs and assumptions

4.1. Threats to Security

The following are the threats that the evaluated product addresses:

T. ACCIDENTAL_ADMIN_ ERROR

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

T.MASQUERADE

A user or process may masquerade as another entity in order to gain unauthorized access to data or TOE resources

T.POOR_DESIGN

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

T.POOR_IMPLEMENTATION

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

T.POOR_TEST

Lack of or insufficient tests to demonstrate that all TOE security functions operate correctly (including in a fielded TOE) may result in incorrect TOE behavior being discovered thereby causing potential security vulnerabilities.

T.RESIDUAL_DATA

A user or process may gain unauthorized access to data through reallocation of TOE resources from one user or process to another.

T.TSF_COMPROMISE

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

T.UNAUTHORIZED_ACCESS

A user may gain unauthorized access to user data for which they are not authorized according to the TOE security policy.

T.UNIDENTIFIED_ACTIONS

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

T.DENIAL_OF_SERVICE

Failure of the master database server might cause the database to become unavailable to users.

4.2. Organizational Security Policies

The following are the Organizational Security Policies of the TOE:

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

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

4.3. Assumptions

The following are the assumptions regarding the security environment and the intended usage of the TOE:

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

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

A.OS_PP_VALIDATED The underlying OS has been validated against an NSA sponsored OS PP of at least Basic Robustness.

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.

5. Architectural Information

Postgres Plus Advanced Server is a relational database system built around a client/server architecture. All relational data managed by the database server is stored in a collection of files that reside in the file system of the host operating system. In this context, host refers to the computer on which the database server resides.

The server does not include a native user interface; all communication between a user and the server is conducted through a client application. Postgres Studio and EDB*Plus (both included in the TOE) are examples of client applications. Before a client application can interact with relational data (reads, writes, or deletes) served by the database server, the client must first establish a network connection to the database server. After the database server has authenticated the client application, the client may send one or more SQL statements across the network connection to the database server. The database server verifies that the client application is authorized to access the requested data, executes the SQL statement, and sends the result back across the network connection to the client application.

5.1.TOE Physical Boundaries

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

- Database Server 8.4.4-400 (in TOE),
- Client Connectors (bundled) (in TOE),
- Postgres Studio 1.10.4 (in TOE),
- PostGIS Spatial Extensions 1.5.1-3 (in TOE),
- EDB*Plus 8.4 (build 25) (in TOE),
- Slony Replication 2.0.3 (in TOE),
- PG Agent (bundled) (in TOE),
- Update Monitor (bundled) (in TOE),
- Infinite Cache Daemon (not in TOE),
- Migration Studio (not in TOE),
- EnterpriseDB Migration Toolkit (not in TOE),
- xDB Replication Server (not in TOE),
- DBA Management Server (not in TOE),
- Monitoring Tools (not in TOE),
- PG Bouncer (not in TOE),
- Procedural Language Debugger (not in TOE)
- StackBuilder Plus (not in TOE)

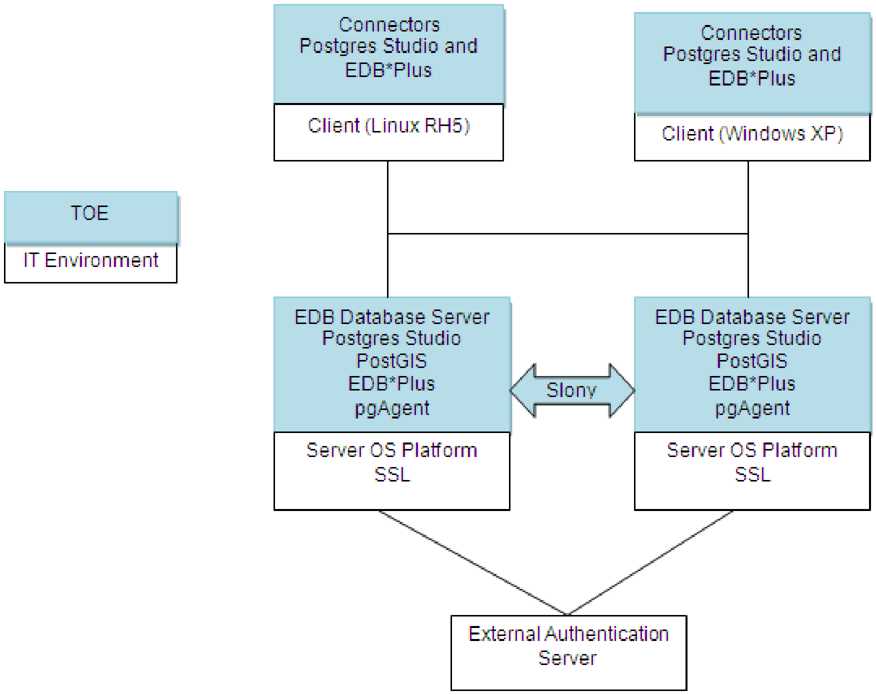

Figure 1: TOE Boundary

The above figure shows a sample configuration with two copies of the Database Server. The figure depicts the physical scope of the TOE within its Operational Environment.

5.2. Clarification of Scope

Configuration Options that are Out of Scope:

- **"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.
- **"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.

5.3.Functional Dependencies on the Operational Environment

The Operational Environment needs to provide the following capabilities:

- 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 Operational Environment (PAM, LDAP, Kerberos, GSSAPI, SSPI, SSL Certificates)
- 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

6. Documentation

6.1.Guidance Documentation

The TOE is delivered to the end user using installer files downloaded from a secured web page (a valid registration account is required for downloading) and documents are available for download in that site as well.

The following documents are developed and maintained by EnterpriseDB Corporation and delivered to the end user of the TOE:

[1] EnterpriseDB Corp, Postgres Plus Advanced Server EAL2 Supplemental Guide, Version 1.1, July 7, 2011

[2] The PostgreSQL Global Development Group; PostgreSQL 8.4.4 Documentation, Version 8.4.4

[3] EnterpriseDB Corp, Postgres Plus Advanced Server Guide, Version 2.1, September 30, 2010

[4] EnterpriseDB Corp, Postgres Plus Advanced Server Oracle Compatibility Developer’s Guide, Version 2.18, September 30, 2010

[5] EnterpriseDB Corp, Postgres Plus Advanced Server Postgres Studio Users Guide, Version 1.0., August 8, 2010

[6] EnterpriseDB Corp, Postgres Plus Advanced Server Installation Guide; Version 1.0, September 30, 2010

[7] EnterpriseDB Corp, Postgres Plus Advanced Server 8.4 ODBC Connector Guide; Version 1.1, September 30, 2010

[8] EnterpriseDB Corp, Postgres Plus Advanced Server 8.4 JDBC Connector Guide; Version 1.2, September 30, 2010

[9] EnterpriseDB Corp, Postgres Plus Advanced Server 8.4 .NET Connector Guide; Version 1.2, August 8, 2010

[10] EnterpriseDB Corp, Postgres Plus Advanced Server 8.4 Performance Features Guide; Version 1.1, September 30, 2010

[11] EnterpriseDB Corp, Tutorial: How to Set Up pgAgent for Postgres Plus; Version 1,February 19, 2010

[12] EnterpriseDB Corp, Tutorial: How to Set Up Slony-I Replication for Postgres Plus; Version 1, February 11, 2010

[13] EnterpriseDB Corp, Tutorial: How to use PostGIS with Postgres Plus Advanced Server; Version 2, June 29, 2010

[14] Refractions Research, Inc., PostGIS 1.5.1

7. IT Product Testing

This section describes the testing efforts of the developer and the evaluation team.

7.1.Developer Testing

The developer testing effort is described in detail in the Developer Test Plan documentation.

7.1.1.Overall Test Approach and Results

Developer testing consisted of the following types of tests:

- Manual Tests that must be performed at the command line
- Manual Tests that must be performed using the graphical client interface (i.e. Postgres Studio Graphical Client)
- Scripted Tests that may be invoked with the Regression Tool (automated tool comparable to the tool used for in house testing of the TOE)
- Scripted Tests that must be invoked and verified manually by the tester/evaluator

Each test within the test suite was designed to be executed against a fresh installation of the TOE, installed to conform to the evaluated configuration of the TOE.

Manual Tests performed at the command line

Each manual test includes the Test case identifier, a reference to the SFR that the case was scripted to satisfy (though the test case may satisfy more than one SFR), a description of the test, and the expected result of the test.

The test description contains a series of steps designed to guide the evaluator through a process that demonstrates SFR support by the TSFI featured in the test.

When possible or appropriate, supporting screenshots that show the anticipated output from the execution of the test have been included in the folder with the test description.

Manual Tests performed using the Postgres Studio Graphical Client

Postgres Studio is a graphical client for the PPAS database server. As a user selects options on a Postgres Studio dialog, Postgres Studio assembles a SQL command. If a Postgres Studio dialog includes a SQL pane, the SQL command built by the user’s selections on the dialog can be viewed by opening the SQL pane. When a user clicks the OK button, the PPAS database server verifies the privileges of the user and executes the SQL command (if the authenticated user has sufficient privileges).

The Postgres Studio Test Evidence document (submitted as part of the ATE_FUN.1 evidence) includes a series of test scenarios and screenshots designed to demonstrate Postgres Studio's support of the SFR enforcement provided by the PPAS server. Each scenario also includes the anticipated output of the test. The Postgres Studio Evidence document also includes screenshots and descriptions that map the fields on the Postgres Studio dialog to the SQL Commands.

Postgres Studio always supports the database server as it enforces any restrictions. For example, if a role is created with a limit on the maximum number of concurrent sessions, or an expiration time, the server will enforce those limits; if a user attempts to exceed those limits, Postgres Studio will display an error message and deny access.

Scripted Tests invoked with the Regression Tool

The scripted test suite exercises TOE interfaces listed in the test matrix (and identified in the FSP evidence). The test suite regression tool can invoke an individual test by name, or a series of related tests (for example, all of the tests that satisfy the testing requirements for a specified SFR). A series of tests is referred to as a 'schedule'.

The output file captures any messages that are returned from a successful execution of the test. As the regression tool exercises a test script, it captures the new output of each script in an 'actual' directory. The regression tool then compares the 'actual' output against the 'expected' output. Any variation between the 'actual' and 'expected' file are suspected to be a failure, and are written to a result file (regression.diffs) for review later. As the regression tool runs, it displays the name of each script and the pass/fail result of the test. When the regression tool completes a schedule, it reports the number of passes and the number of failures.

Scripted Tests invoked and verified manually

Setting up a Slony replication scenario or SQL/Protect test environment is a fairly complex process that involves managing permissions, setting environment variables and other tedious and error-prone steps. To simplify the testing process, the EnterpriseDB developers have created shell scripts that perform the setup and test steps. Each test case within the Slony test suite is in an individual folder, and is accompanied by a README file that instructs the evaluator how to use the test scripts within the suite, and how to confirm that the test has performed successfully.

The Test Summary is a brief description of the behavior of the test scripts; as a test script runs, the conversation between the test scripts and the server is displayed onscreen. The Test Summary can act as a guide to understanding the onscreen text.

Any pre-requisites for the test case are noted before the test steps begin and the test steps direct the evaluator through the process of invoking the test scripts. The test result file (named slony_output.txt) contains the expected output of the test case if the test is successful. Also scripts include 'cleanup' steps which readying the server for the next test.

7.1.2.Depth and Coverage

The developer test plan documentations are designed to demonstrate the SFR-enforcing and supporting behavior of Postgres Plus Advanced Server 8.4 and its components when configured as described in the ST and EAL2 Supplemental Guide. The goal of the test plan is to demonstrate SFR conformance through testing of the server and the integral server components (using 100% the TSFI's identified in the FSP).

The test plan shows that developer has tested 100% of SFR enforcing or supporting behaviors provided by the TOE components. When the interface allows, the tests for the component demonstrates both a positive and negative behavior.

7.1.3.Results

The evaluator checked the test procedures and the test evidence and found that the expected test results are consistent with the actual test results provided. For each test case examined, the evaluator compared the expected results in the test procedures with the actual results `provided in the test evidence and found that the actual results were consistent with the expected results.

Given the Evaluation Assurance level (EAL 2), the evaluator determined that Vendor's TOE testing is adequate. All the external TSF interfaces are tested. TOE testing exercises all security functions identified in the Functional Specification.

7.2.Evaluator Independent Testing

The evaluator performed the following activities during independent testing:

- Execution the Developer's Functional Tests (ATE_IND.2)
- Team-Defined Functional Testing (ATE_IND.2)
- Vulnerability/Penetration Testing (AVA_VAN.2)

7.2.1.Execution the Developer's Functional Tests

The Evaluator's testing strategy was to select test cases that specified complete coverage of all security functions defined in the ST. After the test cases were defined, the EnterpriseDB development team test procedures were used to exercise each test case.

Testing was conducted using VMware Fusion (based on hardware virtualization) that ran Win2003, RHEL5, and Windows XP as guest operating systems. XServe running OS X Server Apple was the host machine.

Postgres Plus Advanced Server v8.4 is software only TOE running as an application on top of the OS (no hardware or appliances are included in the TOE). Also there are no IT requirements relaying directly on the HW. Considering that the virtual HW meets the minimum HW requirements by TOE, using HW virtualization technology will not have any influence or effect in the TOE and/or the TSF.

The sampling of the Developer's Functional Test cases was executed. The TOE was installed in the evaluated configuration consistent with the Security Target. CygnaCom selected approximately 85% of the tests the Developer provided as evaluation evidence. The tests were selected to exercise security functions from the externally visible TSFI. The evaluator ensured that the test sample included the tests such that:

- All Security Functions were tested
- All External interfaces were exercised
- All Security Functional Requirements were tested.

The test configurations used by the evaluator were the same as that used by the developer.

The test results and screenshots for the test cases were recorded during the Evaluator testing. Overall success of the testing was measured by 100% of the retests being consistent with expected results. Anomalies were documented along with suggested / required solutions.

All of the Developer's Functional Tests rerun by the Evaluator received a 'Pass' verdict.

7.2.2. Team-Defined Functional Testing

The Evaluator selected individual test procedures from the set of Developer Functional Tests, and modified the input parameters to ensure fuller coverage of security functions and correctness of developer reported results (ensuring that the results were not canned).

The Evaluation Team's strategy in developing the Team-Defined Functional tests for the TOE was to supplement the Developer Functional tests and the Penetration tests.

The Team-Defined Functional tests are devised to augment the Developer Functional tests in order to exercise functionality in greater depth than the Developer tests provided. In particular, these tests are developed to exercise the primary security functionality of the TOE:

- Revocation (FMT_REV.1)
- Database Server Controlled Switchover/Failover (FPT_OVR_(EXT).1)
- SQL Injection Protection (FPT_SIP_(EXT).1)
- TOE access history (FTA_TAH_(EXT).1) – implemented as the result of this evaluation

The test results and screenshots for the test cases were recorded during the Evaluator testing. Overall success of the testing was measured by 100% of the tests being consistent with expected results. Anomalies were documented along with suggested / required solutions.

All of the Team-Defined Tests received a 'Pass' verdict.

7.3. Vulnerability/Penetration Testing

Testing configuration(s) used for the developer tests and team-defined tests was used for the penetration testing as well.

The penetration tests covered publicly listed vulnerabilities, hypothesized vulnerabilities and potential misuse of guidance. The list of hypothesized vulnerabilities was developed based on the evaluator's analysis of the evaluation evidence for obvious vulnerabilities. The evaluator has considered the following while performing the vulnerability analysis and penetration tests:

- All Evidence Deliverables: All evidence deliverables were considered for identifying potential vulnerabilities. An analysis of the design documentation identified no specific vulnerabilities.

- Public Sources: The evaluator performed independent search for vulnerabilities availably from Public domain including:
  - NVD database (http://web.nvd.nist.gov),
  - CVE (http://cve.mitre.org/cve/), and
  - PostgreSQL Security Information (http://www.postgresql.org/support/security.html).
- TSF based analysis: All security Functions, Security Functional Requirements and External interfaces were considered.
- Subject to Threats: Including Bypass, Tampering, Direct Attacks and Misuse.
- Open Source Scanner: As an additional measure the TOE in its operation was scanned by openVAS equipped with latest Set of Plug-ins.

The test results and screenshots for the test cases were recorded during the evaluator testing. Overall success of this testing was measured by 100% of the tests being consistent with expected results.

The evaluator examined the results of all penetration testing and found that the TOE installed in its intended environment, has no exploitable obvious vulnerabilities.

A test had a “Pass” result if the actual results obtained by the Evaluator when the test was run matched the expected results predicted for the test when it was written by the Evaluation Team prior to testing.

8. Evaluated Configuration

Testing was done using VMware Fusion (based on hardware virtualization) that ran Win2003, RHEL5, and Windows XP as guest operating systems. XServe running OS X Server Apple was the host machine.

Postgres Plus Advanced Server v8.4 is software only TOE running as an application on top of the OS (no hardware or appliances are included in the TOE). In addition, there are no IT requirements relating directly to the HW. Considering that virtual HW meets the minimum HW requirements by TOE, using HW virtualization technology does not have any influence or effect on the TOE and/or the TSF.

The TOE was tested on the following operating system platforms:

- DB Server platforms:
  - 2 Red Hat Linux Version 5
    and
  - 2 Microsoft Windows 2003 Server
- 2 Clients Application platform with all the connectors (JDBC, ODBC, .NET, OCI, and libpq), Postgres Studio and EDB*Plus:
  - 1 MS Windows (XP)
  - 1 Linux (RH5)

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

Operational Environment (outside the scope of this evaluation):

Authenticator servers were not present in the Operational Environment during the testing because there were no tests in the evaluation test plan that required an external Authenticator server(s). The evaluation team tested I&A functions provided wholly within the TOE (Password and MD5 Password authentication).

9. Results of Evaluation

A verdict for an assurance component is determined by the resulting verdicts assigned to the corresponding evaluator action elements. The evaluation was conducted based upon:

- Common Criteria for Information Technology Security Evaluation - Part 2: Security functional components, September 2007 Version 3.1 Revision 2, CCMB-2007-09-002.
- Common Criteria for Information Technology Security Evaluation - Part 3: Security assurance components, September 2007, Version 3.1 Revision 2, CCMB-2007-09-003.
- Common Methodology for Information Technology Security Evaluation - Evaluation methodology, September 2007, Version 3.1 Revision 2, CCMB-2007-09-004.

The Evaluation Team assigned a Pass, Fail, or Inconclusive verdict to each work unit of each EAL 2 augmented by ALC_FLR.2 assurance component. For Fail or Inconclusive work unit verdicts, the Evaluation Team advised the developer of issues requiring resolution or clarification within the evaluation evidence. In this way, the Evaluation Team assigned an overall Pass verdict to the assurance component only when all of the work units for that component had been assigned a Pass verdict.

The details of the evaluation are recorded in the Evaluation Technical Report (ETR), which is controlled by CygnaCom CCTL.

The evaluation team assigned PASS verdicts for all applicable evaluator action elements and consequently all applicable assurance components.

- The TOE is CC Part 2 Extended
- The TOE is CC Part 3 Conformant.

The validators reviewed the findings of the evaluation team, and have concurred that the evidence and documentation of the work performed support the assigned rating.

10. Validators Comments/Recommendations

Note 1: The TOE meets the intent of the PP requirement FAU_GEN.1-NIAP-0410 regarding logging of start-up and shut-down of audit functions by requiring the auditing function to be running all the time. The TOE does not allow for the starting-up and shutting-down of the audit functions while database system is running. The authorized administrator is advised and warned not to modify those parameters that control the auditing functions in the postgresql.conf configuration file. Since an authorized administrator following the guidance documentation will never shutdown the database while the TOE is running, the intent on the requirement is met.

Note 2: According to the developer, PPAS can provide failover/switchover for crossover platforms (Linux ↔ Windows), however, that configuration is not recommended and it is not supported by EnterpriseDB. Therefore the failover/switchover function (provided by Slony TOE component) is been evaluated only for Linux RH5 <-> Linux RH5 and Windows 2003 <-> Windows 2003 configurations.

Note 3: The Syslog and SNMP servers have not been tested as there is no security requirements tied to these interfaces.

11. Security Target

The *EnterpriseDB Postgres Plus Advanced Server v8.4 Security Target*, *Version 1.12, June 2, 2011* is compliant with the Specification of Security Targets requirements found within Annex B of Part 1of the CC.

12. Glossary

12.1. Acronyms

The following are product specific and CC specific acronyms. Not all of these acronyms are used in this document.

API	Application Programming Interface
CC	Common Criteria [for IT Security Evaluation]
CLI	Command Line Interface
DBA	Database Administrator
DDL	Data Definition Language
DBMS	Database Management System
DML	Data Manipulation Language
EAL	Evaluation Assurance Level
FIPS	Federal Information Processing Standards Publication
GSSAPI	Generic Security Services Application Program Interface
HBA	Host-Based Authentication
ID	Identifier
IT	Information Technology
LDAP	Lightweight Directory Access Protocol
NIST	National Institute of Standards and Technology
OCI	Oracle Call Interface
PAM	Pluggable Authentication Modules
PP	Protection Profile
RDBMS	Relational DBMS
SSPI	Security Services Provider Interface
SF	Security Function
SFP	Security Function Policy
SFR	Security Functional Requirements
SPL	Stored Procedure Language
SQL	Structured Query Language
SSL	Secure Socket Layer protocol
ST	Security Target
TOE	Target of Evaluation
TSC	TSF Scope of Control

TSF	TOE Security Functions
TSFI	TOE Security Functions Interface
TSP	TOE Security Policy
UI	User Interface

12.2. Terminology

This section defines the product-specific and CC-specific terms. Not all of these terms are used in this document.

Assignment The specification of an identified parameter in a component.

Assurance Grounds for confidence that an entity meets its security objectives.

Attack potential The perceived potential for success of an attack, should an attack be launched, expressed in terms of a threat agent's expertise, resources and motivation.

Augmentation The addition of one or more assurance component(s) to a package.

Authentication data Information used to verify the claimed identity of a user.

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" apply to all users who have authorized access to the TSF Data. This includes both users with PPAS 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.

Class A grouping of families that share a common focus.

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.

Component The smallest selectable set of elements on which requirements may be based.

Connectivity

The property of the TOE that allows interaction with IT entities external to the TOE. This includes exchange of data by wire or by wireless means, over any distance in any environment or configuration.

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 PPAS. The Superuser only has access to TSF data via TOE interfaces after authentication. The Database Superuser is called the “authorized administrator” in the DBMS PP.

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.

Dependency

A relationship between components such that if a requirement based on the depending component is included in a PP, ST or package, a requirement based on the component that is depended upon must normally also be included in the PP, ST or package..

Element

An indivisible security requirement.

Evaluation

Assessment of a PP, an ST, or a TOE against defined criteria.

Evaluation Assurance Level (EAL)

A package consisting of assurance components from Part 3 that represents a point on the CC predefined assurance scale.

Evaluation authority

A body that implements the CC for a specific community by means of an evaluation scheme and thereby sets the standards and monitors the quality of evaluations conducted community.

Evaluation scheme

The administrative and regulatory framework under which the CC is applied by an evaluation authority within a specific community.

Extension

The addition to an ST or PP of functional requirements not contained in Part 2 and/or assurance requirements not contained in Part 3 of the CC.

External entity

Any entity (human or IT) outside the TOE that interacts (or may interact) with the TOE.

Family

A grouping of components that share security objectives but may differ in emphasis or rigor.

Formal

Expressed in a restricted syntax language with defined semantics based on well-established mathematical concepts.

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.

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.

Informal

Expressed in natural language.

Inter-TSF transfers

Communicating data between the TOE and the security functions of other trusted IT products.

Internal communication channel

A communication channel between separated parts of TOE.

Internal TOE transfer

Communicating data between separated parts of the TOE.

Iteration

The use of the same component to express two or more distinct requirements.

Object

A passive entity in the TOE, that contains or receives information, and upon which subjects perform operations.

Organizational security policies

A set of security rules, procedures, or guidelines imposed (or presumed to be imposed) now and/or in the future by an actual or hypothetical organisation in the operational environment.

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.

Protection Profile (PP)

An implementation-independent statement of security needs for a TOE type.

Prove

This term refers to a formal analysis in its mathematical sense. It is completely rigorous in all ways. Typically, “prove” is used when there is a desire to show correspondence between two TSF representations at a high level of rigor.

Refinement

The addition of details to a component.

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.

Secret

Information that must be known only to authorized users and/or the TSF in order to enforce a specific SFP.

Secure state

A state in which the TSF data are consistent and the TSF continues correct enforcement of the SFRs.

Security attribute

A property of subjects, users (including external IT products), objects, information, sessions and/or resources that is used in defining the SFRs and whose values are used in enforcing the SFRs.

Security Function Policy (SFP)
A set of rules describing specific security behaviour enforced by the TSF and expressible as a set of SFRs.

Security objective
A statement of intent to counter identified threats and/or satisfy identified organisation security policies and/or assumptions.

Security Target (ST)
An implementation-dependent statement of security needs for a specific identified TOE.

Selection
The specification of one or more items from a list in a component.

Semiformal
Expressed in a restricted syntax language with defined semantics.

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

Subject
An active entity in the TOE that performs operations on objects.

Target of Evaluation (TOE)
A set of software, firmware and/or hardware possibly accompanied by guidance.

TOE resource
Anything useable or consumable in the TOE.

TOE Security Functions (TSF)
A set consisting of all hardware, software, and firmware of the TOE that must be relied upon for the correct enforcement of the TSP.

Transfers outside TSF
TSF mediated communication of data to entities not under control of the TSF.

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.

Trusted channel
A means by which a TSF and a remote trusted IT product can communicate with necessary confidence.

Trusted path
a means by which a user and a TSF can communicate with necessary confidence.

TSF data
Data created by and for the TOE that might affect the operation of the TOE.

TSF interface (TSFI)
A means by which external entities (or subjects in the TOE but outside of the TSF) supply data to the

TSF, receive data from the TSF and invoke services from the TSF.

User — See external entity

User data — Data created by and for the user that does not affect the operation of the TSF.

User or Advanced Server user — In PPAS, the term “user” refers to an entity representing an individual as in many other IT systems. However, a “user” in PPAS is implemented as a role that has been granted the LOGIN privilege.

13. Bibliography

[1] Common Criteria for Information Technology Security Evaluation - Part 1: Introduction and general model, September 2006 Version 3.1 Revision 1, CCMB-2006-09-001.

[2] Common Criteria for Information Technology Security Evaluation - Part 2: Security functional components, September 2007 Version 3.1 Revision 2, CCMB-2007-09-002.

[3] Common Criteria for Information Technology Security Evaluation - Part 3: Security assurance components, September 2007, Version 3.1 Revision 2, CCMB-2007-09-003.

[4] Common Methodology for Information Technology Security Evaluation - Evaluation methodology, September 2007, Version 3.1 Revision 2, CCMB-2007-09-004.

[5] Common Criteria Evaluation and Validation Scheme (CCEVS): (http://www.niap-ccevs.org/cc-scheme).

[6] CygnaCom Solutions CCTL (http://www.cygnacom.com).