Microsoft SQL Server™ 2005 Database Engine Common Criteria Evaluation

Security Target

Microsoft SQL Server Team

Author:	Roger French
Version:	1.27
Date	2008-07-23
File Name:	MS_SQL_ST_1.27

Abstract

This document is the Security Target (ST) for the Common Criteria evaluation of the Database Engine of Microsoft SQL Server™ 2005, Enterprise Edition (English), Version 9.00.3068.00 (TOE)

Keywords

CC, ST, Common Criteria, SQL, Security Target

Revision History

Date	Version	Author	Edit
2005-10-25	0.1	Roger French	Initial Version
2005-10-26	0.2	Roger French	First content for TOE description
2005-10-29	0.3	Roger French	Content of DBMS-PP V.1.21 incorporated (without Rationale)
2005-11-06	0.4	Roger French	Rationale incorporated, formatting
2005-11-13	0.45	Roger French	First draft of the TOE summary specification added
2005-11-14	0.5	Roger French	Updated the TOE description and added some information to the Rationale
2005-11-15	0.6	Roger French	Reviewed the SFRs, performed the open operations, editing of the TOE Summary Specification
2005-11-21	0.7	Roger French	Complete review for the first draft version
2005-11-24	0.71	Roger French	Editorial changes
2005-12-19	0.75	Roger French	Changes after Kick-Off Meeting with BSI
2005-12-19	0.78	Roger French	Removed references to PP
2005-12-21	0.79	Roger French	Editorial changes
2005-12-22	0.8	Roger French	Editorial Changes
2005-12-23	0.81	Roger French	Editorial Changes
2005-12-23	0.82	Roger French	Editorial Changes
2005-12-27	0.83	Roger French	Editorial Changes
2005-12-28	0.84	Roger French	Editorial Changes
2006-01-03	0.85	Roger French	Editorial Changes
2006-01-20	0.86	Roger French	Changes after 1st OR
2006-01-20	0.9	Roger French	Editorial Changes, Version for 2nd review by the evaluator
2006-01-24	0.91	Roger French	Editorial Changes
2006-01-25	0.92	Roger French	Changed description of instances and changed SFR for the environment
2006-01-26	0.93	Roger French	Changed FIA_ATD.1, FDP_ACF.1.3
2006-01-30	0.96	Roger French	Final version after evaluation
2006-03-15	0.97	Roger French	Addressed comments of BSI
2006-03-20	0.98	Roger French	Minor changes to Management function
2006-04-18	0.99	Roger French	Minor changes regarding transmission of password
2006-05-22	1.0	Roger French	Added statement in chapter 1.2
2006-07-09	1.01	Roger French	Minor changes to the rationale and chapter 5
2006-07-12	1.1	Roger French	Updated ST to be compliant to [PP]
2006-08-08	1.11	Roger French	Minor editorial changes
2006-08-15	1.12	Roger French	Using FDP_RIP.2 instead of FDP_RIP.1
2006-09-08	1.13	Roger French	Minor editorial changes
2006-09-11	1.14	Roger French	Minor editorial changes
2007-04-03	1.15	Roger French	Incorporated comments from EAL1 evaluation
2007-05-15	1.16	Roger French	Updated default value for # of concurrent sessions

2007-08-03	1.17	Roger French	Minor updates
2007-08-30	1.18	Roger French	Updated permissions of database roles
2007-09-03	1.19	Roger French	Update final build number
2007-10-22	1.2	Roger French	Minor update in SF.AU
2007-10-22	1.21	Roger French	Minor update after evaluators comments
2008-02-12	1.22	Roger French	Minor update after evaluators comments
2008-02-16	1.23	Roger French	Minor update after evaluators comments
2008-02-20	1.25	Roger French	Minor update after evaluators comments
2008-03-24	1.26	Roger French	Minor update
2008-07-22	1.27	Roger French	Final version

This page intentionally left blank

Table of Contents

List of Tables

List of Figures

1 ST Introduction

This chapter presents security target (ST) identification information and an overview of the ST. An ST contains the information technology (IT) security requirements of an identified Target of Evaluation (TOE) and specifies the functional and assurance security measures offered by that TOE to meet stated requirements. An ST principally defines:

a) A security problem expressed as a set of assumptions about the security aspects of the environment, a list of threats that the TOE is intended to counter, and any known rules with which the TOE must comply (chapter 3, TOE Security Environment).

b) A set of security objectives and a set of security requirements to address the security problem (chapters 4 and 5, Security Objectives and IT Security Requirements, respectively).

c) The IT security functions provided by the TOE that meet the set of requirements (chapter 6, TOE Summary Specification).

The structure and content of this ST comply with the requirements specified in the Common Criteria (CC), Part 1, Annex C, and Part 3, chapter 5.

1.1 ST Identification

This chapter provides information needed to identify and control this ST and its Target of Evaluation (TOE).

ST Title:	**Microsoft SQL Server 2005 Database Engine Common Criteria Evaluation Security Target**
ST Version:	1.27
Date:	2008-07-23
Author:	Roger French, Microsoft Corporation
Certification-ID	BSI-DSZ-CC-366
TOE Identification:	Database Engine of Microsoft SQL Server 2005, Enterprise Edition (English), and its related guidance documentation.
TOE Version:	9.00.3068.00[1]
TOE Platform:	Windows Server 2003 Enterprise Edition (English) SP1 including MS05-042, MS05-039, MS05-027, A patch that updates the Internet Protocol (IP) Security (IPSec) Policy Agent is available for Windows Server 2003 and Windows XP (KB 907865) as specified in [WIN_ST].
CC Identification:	Common Criteria for Information Technology Security Evaluation, Version 2.3, August 2005 (also known as ISO 15408).
Evaluation Assurance Level:	EAL4 augmented by ALC_FLR.2
PP Conformance:	U.S. Government Protection Profile for Database Management Systems in Basic Robustness Environments, Version 1.1, 07.06.2006
Keywords:	CC, ST, Common Criteria, SQL, Security Target

1.2 ST Overview

The TOE is the database engine of SQL Server 2005. SQL Server is a Database Management System (DBMS).

The TOE has been developed as the core of the DBMS to store data in a secure way.

The security functionality of the TOE comprises:

- Security Management
- Access Control
- Identification and Authentication
- Security Audit
- Session Handling

[1] This version includes the Service Pack 2 (SP2) and the Security Patch GDR 4

A summary of the TOE security functions can be found in chapter 2, TOE Description. A more detailed description of the security functions can be found in chapter 6, TOE Summary Specification.

Please note that only the SQL Server 2005 database engine is addressed in this ST. Other related products of the SQL Server 2005 platform, such as Service Broker, provide services that are useful but are not central to the enforcement of security policies. Hence, security evaluation is not directly applicable to those other products.

1.3 CC Conformance

The TOE is [CC_PART2] extended and [CC_PART3] conformant at the level of assurance EAL4 augmented by assurance requirement ALC_FLR.2.

1.4 Conventions

For this Security Target corresponding to [PP] the following conventions are used:

The CC allows several operations to be performed on functional requirements; *refinement*, *selection*, *assignment*, and *iteration* are defined in paragraph 148 of Part 1 of the CC. Each of these operations is used in this ST.

The **refinement** operation is used to add detail to a requirement, and thus further restricts a requirement. Refinement of security requirements is denoted by **bold text**.

The **selection** operation is used to select one or more options provided by the CC in stating a requirement. Selections that have been made are denoted by *italicized text*.

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 are denoted by showing the value in square brackets, [Assignment_value].

The **iteration** operation is used when a component is repeated with varying operations. Iteration is denoted by showing the iteration number in parenthesis following the component identifier, (iteration_number).

National Information Assurance Partnership (NIAP) interpretations are used and are presented with the NIAP interpretation number as part of the requirement identifier (e.g., **FAU_GEN.1- NIAP-0410** for Audit data generation).

The CC paradigm also allows protection profile and security target authors to create their own requirements. Such requirements are termed ‘explicit requirements’ and are permitted if the CC does not offer suitable requirements to meet the authors’ needs. **Explicit requirements** must be identified and are required to use the CC class/family/component model in articulating the requirements. In this ST, explicit requirements will be indicated with the “_EXP” following the component name.

This ST also includes security requirements on the IT environment. Explicit Environmental requirements will be indicated with the “_(ENV)” following the component name.

Additionally ***bold and italics*** text is used to indicate where text from [PP] has been added or changed as denoted in each chapter of this document.

2 TOE Description

This chapter provides context for the TOE evaluation by identifying the product type and describing the evaluated configuration. The main purpose of this chapter is to bind the TOE in physical and logical terms. The chapter starts with a description of the product type before it introduces the physical scope, the architecture and last but not least the logical scope of the TOE.

2.1 Product Type

The product type of the Target of Evaluation (TOE) described in this ST is a database management system (DBMS) with the capability to limit TOE access to authorized users, enforce Discretionary Access Controls on objects under the control of the database management system based on user and/or role authorizations, and to provide user accountability via audit of users' actions.

A DBMS is a computerized repository that stores information and allows authorized users to retrieve and update that information. A DBMS may be a single-user system, in which only one user may access the DBMS at a given time, or a multi-user system, in which many users may access the DBMS simultaneously.

The TOE which is described in this ST is the database engine and therefore part of SQL Server 2005. It provides a relational database engine providing mechanisms for Access Control, Identification and Authentication, Security Audit and Session handling.

SQL Server additionally includes the following tools which are not part of the TOE:

- Replication Services: Data replication for distributed or mobile data processing applications and integration with heterogeneous systems, including existing Oracle databases.
- Notification Services: Notification capabilities for the development and deployment of applications that can deliver personalized, timely information updates to a variety of connected and mobile devices.
- Integration Services: Extract, transform, and load capabilities for data warehousing and enterprise-wide data integration
- Analysis Services: Online analytical processing (OLAP) capabilities for the analysis of large and complex datasets.
- Reporting Services: A comprehensive solution for creating, managing, and delivering both traditional, paper-oriented reports and interactive, Web-based reports.
- Management tools: SQL Server includes integrated management tools for database management and tuning as well as tight integration with tools such as Microsoft Operations Manager (MOM) and Microsoft Systems Management Server (SMS). Standard data access protocols drastically reduce the time it takes to integrate data in SQL Server with existing systems. In addition, native Web service support is built into SQL Server to ensure interoperability with other applications and platforms.

- Development tools: SQL Server offers integrated development tools for the database engine, data extraction, transformation, and loading (ETL), data mining, OLAP, and reporting that are tightly integrated with Microsoft Visual Studio to provide end-to-end application development capabilities. Every major subsystem in SQL Server ships with it's own object model and set of APIs to extend the data system in any direction that is unique to each business.

The TOE itself only comprises the database engine of the SQL Server 2005 platform which provides the security functionality as required by this ST. All the additional tools as listed before interact with the TOE as a standard SQL client. The scope and boundary of the TOE will be described in the next chapter.

2.2 Physical Scope and Boundary of the TOE

The TOE is the database engine of the SQL Server 2005 and its related guidance documentation.

The following figure shows the TOE (including its internal structure) and its immediate environment.

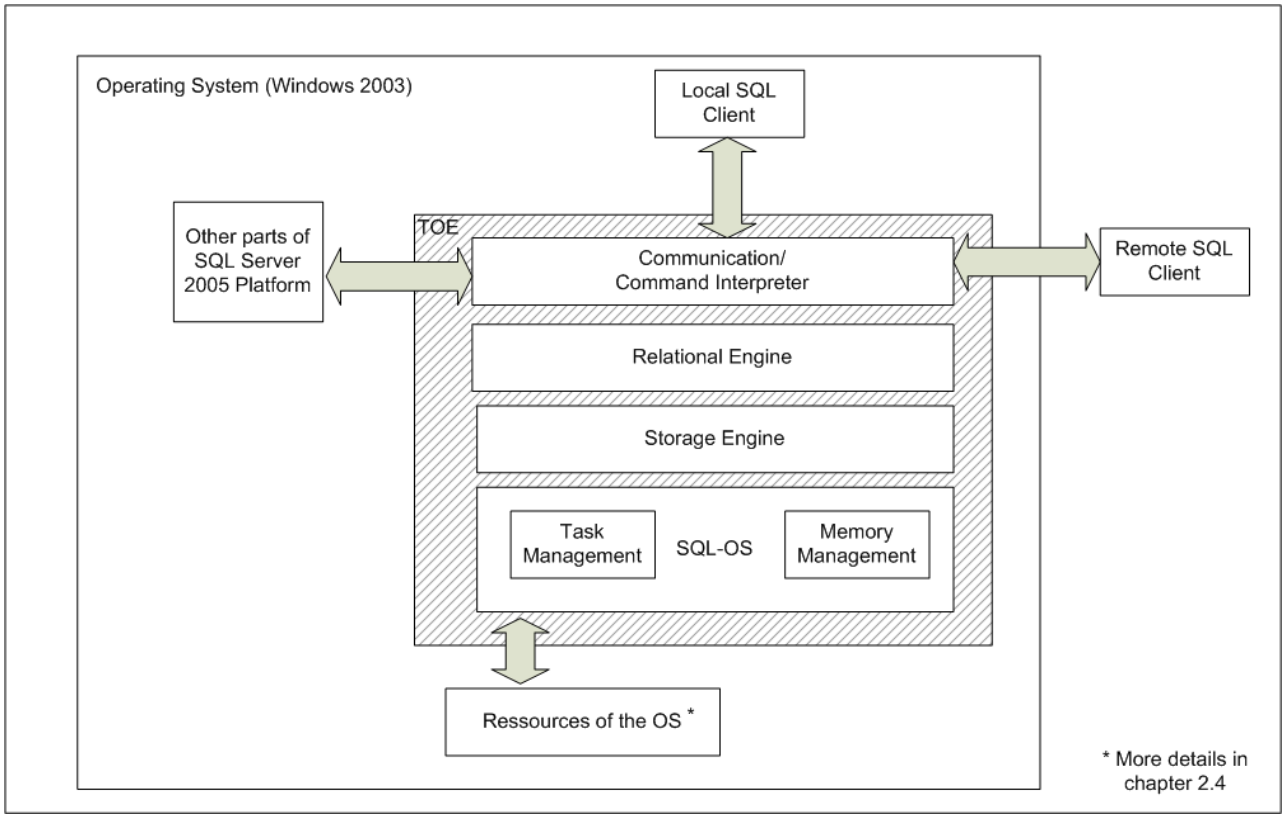

Figure 1: TOE

As seen in figure 1 the TOE internally comprises the following logical units:

The **Communication** part is the interface for programs accessing the TOE. It is the interface between the TOE and clients performing requests. It processes Tabular Data Stream (TDS)

packets to identify the type of packet and translate the packet type into a specific request type.

All responses to user application requests return to the client through this part of the TOE.

The **Relational Engine** is the core of the database engine and is responsible for all security relevant decisions. The relational engine establishes a user context, syntactically checks every Transact SQL (T-SQL) statement, compiles every statement, checks permissions to determine if the statement can be executed by the user associated with the request, optimizes the query request, builds and caches a query plan, and executes the statement.

The **Storage Engine** is a resource provider. When the relational engine attempts to execute a T-SQL statement that accesses an object for the first time, it calls upon the storage engine to retrieve the object, put it into memory and return a pointer to the execution engine. To perform these tasks, the storage engine manages the physical resources for the TOE by using the Windows OS.

The **SQL-OS** is a resource provider for all situations where the TOE uses functionality of the operating system. SQL-OS provides an abstraction layer over common OS functions and was designed to reduce the number of context switches within the TOE. SQL-OS especially contains functionality for Task Management and for Memory Management.

For **Task Management** the TOE provides an OS-like environment for threads, including scheduling, and synchronization —all running in user mode, all (except for I/O) without calling the Windows Operating System.

The **Memory Manager** is responsible for the TOE memory pool. The memory pool is used to supply the TOE with its memory while it is executing. Almost all data structures that use memory in the TOE are allocated in the memory pool. The memory pool also provides resources for transaction logging and data buffers.

The immediate **environment** of the TOE comprises:

The Windows 2003 Server Enterprise Edition Operating System, which hosts the TOE. As the TOE is a software only TOE it lives as a process in the Operating System (OS) and uses the resources of the OS. These resources comprise general functionality (e.g. the memory management and scheduling features of the OS) as well as specific functionality of the OS, which is important for the Security Functions of the TOE (see chapter 2.4for more details)

Other parts of the SQL Server 2005 Platform, which might be installed together with the TOE. The TOE is the central part of a complete DBMS platform, which realizes all Security Functions as described in this ST. However other parts of the platform may be installed on the same machine if they are needed to support the operation or administration of the TOE. However these other parts will interact with the TOE in the same way, every other client would do.

Clients comprising (local clients and remote clients) are used to interact with the TOE during administration and operation. Services of the Operating System are used to route the communication of remote clients with the TOE.

The TOE relies on functionality of the Windows 2003 Server Operating System and has the following hardware requirements:

- 600-megahertz (MHz) Pentium III-compatible or faster processor; 1-gigahertz (GHz) or faster processor recommended
- 512 megabytes (MB) of RAM or more; 1 gigabyte (GB) or more recommended
- Approximately 350 MB of available hard-disk space for the recommended installation
- Approximately 425 MB of additional available hard-disk space for SQL Server Books Online, SQL Server Mobile Books Online, and sample databases
- CD-ROM or DVD-ROM drive
- Super VGA (1,024x768) or higher-resolution video adapter and monitor
- Microsoft Mouse or compatible pointing device

The following guidance documents and supportive information belong to the TOE:

- SQL Server Books Online, February 2007
- SQL Server Guidance Addendum / Installation / Startup

The website https://www.microsoft.com/sql/commoncriteria/2005/sp2/default.mspx provides the guidance documentation and contains additional information about the TOE and its evaluated configuration. This website shall be visited before using the TOE.

2.3 Architecture of the TOE

The TOE which is described in this ST comprises one instance of the SQL-Server 2005 database engine but has the possibility to serve several clients simultaneously. All clients which connect to the TOE are within the same enclave as the TOE which means that they are under the same management control and operate under the same security policy constraints.

2.4 Logical Scope and Boundary of the TOE

SQL Server 2005 is able to run multiple instances of the database engine on one machine. After installation one default instance exists. However the administrator is able to add more instances of SQL Server 2005 to the same machine.

The TOE comprises one instance of SQL Server 2005. Within this ST it is referenced either as "the TOE" or as "instance". The machine the instances are running on is referenced as "server" or "DBMS-server".

If more than one instance of SQL Server 2005 is installed on one machine these just represent multiple TOEs as there is no other interface between two instances of the TOE than the standard client interface

In this way two or more instances of the TOE may only communicate through the standard client interface.

The TOE provides the following set of security functionality

- The **Access Control** function of the TOE ensures that only authorized users are able to connect to the TOE and access user data stored in the TOE. It further controls that only authorized administrators are able to manage the TOE.
- The **Security Audit function** of the TOE produces log files about all security relevant events.
- The **Management** function allows authorized administrators to manage the behavior of the security functions of the TOE.
- The **Identification and Authentication**[2] function of the TOE is able to identify and authenticate users based on a Username/Password based mechanism.
- The **Session Handling** mechanism which limits the possibilities of users to establish sessions with the TOE and maintains a separate execution context for every operation.
- The **Memory Management** functionality of the TOE ensures that any previous information in memory is made unavailable before the memory is used either by overwriting the memory explicitly with a certain pattern or by overwriting the memory

[2] Note that the TOE as well as the environment provides a mechanism for identification and authentication. Chapter 6 will describe this in more detail.

completely with new information.

The following functions are part of the environment:

- The **Audit Review** and **Audit Storage** functionality has to be provided by the environment and provide the authorized administrators with the capability to review the security relevant events of the TOE.
- The **Access Control Mechanisms** has to be provided by the environment for files stored in the environment
- The environment provides **Identification and Authentication** for users for the cases where this is required by the TOE (The environment AND the TOE provide mechanisms for user authentication. See chapter 6.1.3 for more details).
- The environment has to ensure that the security functions of the TOE cannot be **bypassed**.
- The environment has to provide a mechanism for **Domain Separation** to separate the execution context of the TOE from other contexts and to provide the TOE with the capability to create separate contexts for different users.
- The environment provides a **cryptographic** mechanisms for **hashing** of passwords
- The environment provides **residual information protection** for memory which is allocated to the TOE:

All these functions are provided by the underlying Operating System (Windows 2003 Server Enterprise Edition) except Audit Review, for which an additional tool has to be used (e.g. the SQL Server Profiler, which is part of the SQL Server Platform).

Access to the complete functionality of the TOE is possible via a set of SQL-commands (see [TSQL]).

This set of commands is available via:

- Shared Memory
- Named Pipes
- TCP/IP

3 TOE Security Environment

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

3.1 Assets

The TOE maintains two types of data which represent the assets: User Data and TSF Data.

The primary assets are the User Data which comprises the following:

- The user data stored in or as database objects;
- User-developed queries or procedures that the DBMS maintains for users.

The secondary assets comprise the TSF data that the TOE maintains and uses for its own operation. This kind of data is also called metadata. It especially includes:

- The definitions of user databases and database objects
- Configuration parameters,
- User security attributes,
- Transaction logs,
- Security Audit instructions and records

3.2 Assumptions

The following table lists all the assumptions about the environment of the TOE. These assumptions have been taken from [PP] with only the changes indicated by bold italic text.

Table 1 - Assumptions

Assumption	Description
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. ***The evaluation and certification of the underlying OS has been done on at least EAL 4 augmented by ALC_FLR.2.***
A.PHYSICAL	It is assumed that appropriate physical security is provided within the domain for the value of the IT assets protected by the TOE and the value of the stored, processed, and transmitted information.
A.COMM	***It is assumed that any communication path from and to the TOE is appropriately secured to avoid eavesdropping and manipulation.***

The assumption A.OS_PP_VALIDATED ensures that the following security functionality is provided by the environment:

- Identification and authentication of users
- Access Control for Files
- Domain Separation
- Non Bypassability of TOE Security Functions
- Cryptographic Functionality
- Residual Information Protection
- Audit Review and Audit Storage
- Time Stamps

3.3 Threats

3.3.1 Threat Agent Characterization

In addition to helping define the robustness appropriate for a given environment, the threat agent is a key component of the formal threat statements in the ST. Threat agents are typically characterized by a number of factors such as *expertise*, *available resources*, and *motivation*. Because each robustness level is associated with a variety of environments, there are corresponding varieties of specific threat agents (that is, the threat agents will have different combinations of motivation, expertise, and available resources) that are valid for a given level of robustness. The following discussion explores the impact of each of the threat agent factors on the ability of the TOE to protect itself (that is, the robustness required of the TOE).

The *motivation* of the threat agent seems to be the primary factor of the three characteristics of threat agents outlined above. Given the same expertise and set of resources, an attacker with low motivation may not be as likely to attempt to compromise the TOE. For example, an entity with no authorization to low value data none-the-less has low motivation to compromise the data; thus a basic robustness TOE should offer sufficient protection.

Likewise, the fully authorized user with access to highly valued data similarly has low motivation to attempt to compromise the data, thus again a basic robustness TOE should be sufficient.

Unlike the motivation factor, however, the same can't be said for *expertise*. A threat agent with low motivation and low expertise is just as unlikely to attempt to compromise a TOE as an attacker with low motivation and high expertise; this is because the attacker with high expertise does not have the motivation to compromise the TOE even though they may have the expertise to do so. The same argument can be made for *resources* as well.

Therefore, when assessing the robustness needed for a TOE, the motivation of threat agents should be considered a “high water mark”. That is, the robustness of the TOE should increase as the motivation of the threat agents increases.

Having said that, the relationship between expertise and resources is somewhat more complicated. In general, if resources include factors other than just raw processing power (money, for example), then expertise should be considered to be at the same “level” (low, medium, high, for example) as the resources because money can be used to purchase expertise. Expertise in some ways is different, because expertise in and of itself does not automatically procure resources. However, it may be plausible that someone with high expertise can procure the requisite amount of resources by virtue of that expertise (for example, hacking into a bank to obtain money in order to obtain other resources).

It may not make sense to distinguish between these two factors; in general, it appears that the only effect these may have is to lower the robustness requirements. For instance, suppose an organization determines that, because of the value of the resources processed by the TOE and the trustworthiness of the entities that can access the TOE, the motivation of those entities would be “medium”. This normally indicates that a medium robustness TOE

would be required because the likelihood that those entities would attempt to compromise the TOE to get at those resources is in the “medium” range. However, now suppose the organization determines that the entities (threat agents) that are the least trustworthy have no resources and are unsophisticated. In this case, even though those threat agents have medium motivation, the likelihood that they would be able to mount a successful attack on the TOE would be low, and so a basic robustness TOE may be sufficient to counter that threat.

It should be clear from this discussion that there is no “cookbook” or mathematical answer to the question of how to specify exactly the level of motivation, the amount of resources, and the degree of expertise for a threat agent so that the robustness level of TOEs facing those threat agents can be rigorously determined. However, an organization can look at combinations of these factors and obtain a good understanding of the likelihood of a successful attack being attempted against the TOE. Each organization wishing to procure a TOE must look at the threat factors applicable to their environment; discuss the issues raised in the previous paragraph; consult with appropriate accreditation authorities for input; and document their decision regarding likely threat agents in their environment.

The important general points we can make are:

- The motivation for the threat agent defines the upper bound with respect to the level of robustness required for the TOE.
- A threat agent’s expertise and/or resources that is “lower” than the threat agent’s motivation (e.g., a threat agent with high motivation but little expertise and few resources) may lessen the robustness requirements for the TOE (see next point, however).
- The availability of attacks associated with high expertise and/or high availability of resources (for example, via the Internet or “hacker chat rooms”) introduces a problem when trying to define the expertise of, or resources available to, a threat agent.

For this Security Target the attack potential is considered to be low.

The following table identifies the threats to the TOE. These threats have been directly taken from [PP] with only the changes indicated by bold italic text.

Table 2 - Threats to the TOE

Threat	Description
T. ACCIDENTAL_ADMIN_ERROR	An administrator may incorrectly install or configure the TOE resulting in ineffective security mechanisms.
T.MASQUERADE	A user or process may masquerade as another entity in order to gain unauthorized access to data or TOE resources
T.POOR_DESIGN	***Errors*** in requirements specification or design of the TOE may occur, leading to flaws that

Threat	Description
	may be exploited by ***an attacker.***
T.POOR_IMPLEMENTATION	***Errors*** in implementation of the TOE design may occur, leading to flaws that may be exploited by ***an attacker.***
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 authorized administrators to identify and act upon unauthorized actions may occur.

3.4 Organizational Security Policies

An organizational security policy is a set of rules, practices, and procedures imposed by an organization to address its security needs. This chapter identifies the organizational security policies applicable to the TOE. These organizational security policies have been taken from [PP] without any changes.

Table 3 – Organizational Security Policies

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

4 Security Objectives

The purpose of the security objectives is to detail the planned response to a security problem or threat. Threats can be directed against the TOE or the security environment or both therefore, the CC identifies two categories of security objectives:

- Security objectives for the TOE, and
- Security objectives for the environment.

4.1 Security Objectives for the TOE

This chapter identifies and describes the security objectives of the TOE. The objectives have been directly taken from [PP] with only the changes indicated by bold italic text.

Table 4 - Security Objectives for the TOE

Objective	Description
O.ACCESS_HISTORY	The TOE will store and retrieve information (to authorized users) related to previous attempts to establish a session.
O.ADMIN_GUIDANCE	The TOE will provide administrators with the necessary information for secure management.
O.ADMIN_ROLE	The TOE will provide authorized administrators roles to isolate administrative actions.
O.AUDIT_GENERATION	The TOE will provide the capability to detect and create records of security relevant events associated with users.
O.CONFIGURATION_IDENTIFICATION	The configuration of the TOE is fully identified in a manner that will allow implementation errors to be identified and corrected with the TOE being redistributed promptly.
O.DOCUMENTED_DESIGN	The design of the TOE is adequately and accurately documented.
O.INTERNAL_TOE_DOMAINS	The TSF will maintain internal domains for separation of data and queries belonging to concurrent users.

Objective	Description
O.MANAGE	The TOE will provide all the functions and facilities necessary to support the authorized administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.
O.MEDIATE	The TOE must protect user data in accordance with its security policy.
O.PARTIAL_FUNCTIONAL_TEST	The TOE will undergo some security functional testing that demonstrates the TSF satisfies some of its security functional requirements.
O.PARTIAL_SELF_PROTECTION	The TSF will maintain a domain for its own execution that protects itself and its resources from external interference, tampering, or unauthorized disclosure through its own interfaces.
O.RESIDUAL_INFORMATION	The TOE will ensure that any information contained in a protected resource within its Scope of Control is not released when the resource is reallocated.
O.TOE_ACCESS	The TOE will provide mechanisms that control a user's logical access to the TOE.
O.VULNERABILITY_ANALYSIS	The TOE will undergo some vulnerability analysis to demonstrate that the design and implementation of the TOE does not contain any ***flaws which can be exploited in the intended environment for the TOE and that the TOE is resistant to penetration attacks performed by attackers possessing a low attack potential.***
O.I&A	***The TOE will provide a mechanism for identification and authentication of users.***

4.2 Security Objectives for the Environment

The security objectives for the TOE Environment are defined in the following table. The objectives for the environment have been directly taken from [PP] with only the changes indicated by bold italic text.

Table 5 - Security Objectives for the TOE Environment

Objective	Description
OE.NO_EVIL	Sites using the TOE shall ensure that authorized administrators are non-hostile, appropriately trained and follow all administrator guidance.
OE.NO_GENERAL_ PURPOSE	There will be no general-purpose computing capabilities (e.g., compilers or user applications) available on DBMS servers, other than those services necessary for the operation, administration and support of the DBMS.
OE.OS_PP_VALIDATED	The underlying OS has been validated against an NSA sponsored OS PP of at least Basic Robustness. ***The evaluation and certification of the underlying OS has to be done on at least EAL 4 augmented by ALC_FLR.2.***
OE.PHYSICAL	Physical security will be provided within the domain for the value of the IT assets protected by the TOE and the value of the stored, processed, and transmitted information.
OE.COMM	***Any communication path from and to the TOE will be appropriately secured to avoid eavesdropping and manipulation.***

5 IT Security Requirements

This chapter defines the IT security requirements that shall be satisfied by the TOE or its environment:

The CC divides TOE security requirements into two categories:

- Security functional requirements (SFRs) (such as, identification and authentication, security management, and user data protection) that the TOE and the supporting evidence need to satisfy to meet the security objectives of the TOE.
- Security assurance requirements (SARs) that provide grounds for confidence that the TOE and its supporting IT environment meet its security objectives (e.g., configuration management, testing, and vulnerability assessment).

These requirements are discussed separately within the following subchapters.

5.1 TOE Security Functional Requirements

The TOE satisfies the SFRs delineated in the following table. The rest of this chapter contains a description of each component and any related dependencies.

Table 6 - TOE Security Functional Requirements

Class FAU: Security Audit	
FAU_GEN.1-NIAP-0410	Audit data generation
FAU_GEN_EXP.2	User and/or group identity association
FAU_SEL.1-NIAP-0407	Selective audit
FAU_STG_EXP.4	Administrable Prevention of audit data loss
Class FDP: User Data Protection	
FDP_ACC.1	Subset access control
FDP_ACF.1-NIAP-0407	Security attribute based access control
FDP_RIP.***2***	***Full*** Residual Information Protection
Class FIA: Identification and Authentication	
FIA_ATD.1	User attribute definition
FIA_UAU.2	User authentication before any action
FIA_UAU.5	Multiple authentication mechanisms
FIA_UID.2	User identification before any action
Class FMT: Security Management	
FMT_MOF.1	Management of security functions behaviour
FMT_MSA.1	Management of security attributes
FMT_MSA_EXP.3	Static attribute initialization
FMT_MTD.1	Management of TSF data

FMT_REV.1(1)	Revocation (user attributes)
FMT_REV.1(2)	Revocation (subject, object attributes)
FMT_SMF.1	Specification of management functions
FMT_SMR.1	Security roles
Class FPT: Protection of the TSF	
FPT_SEP_EXP.1	TSF domain separation
FPT_TRC_EXP.1	Internal TSF consistency
Class FTA: TOE access	
FTA_MCS.1	Basic limitation on multiple concurrent sessions
FTA_TAH_EXP.1	TOE access history
FTA_TSE.1	TOE session establishment

5.1.1 Class FAU: Security Audit

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

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 7**[3];

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

d) Use of special permissions (e.g., those often used by authorized administrators[4] to circumvent access control policies); and

e) [*events as specified in Table 8*]].

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 ST, [information specified in column three of Table 7 and 8 below].

Table 7 - Auditable Events

[3] Number of table changed by ST writer.

[4] Note that in the context of this Security Target the term „Authorized Administrator“ refers either to the „sysadmin“ (sa) or any other user who has the permission to perform the administration activity based on the DAC policy (see also chapter 9.4.1).

Security Functional Requirement	Auditable Event(s)	Additional Audit Record Content
FAU_GEN.1-NIAP-0410	None	-
FAU_GEN_EXP.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.***2***	None	
FIA_ATD.1	None	-
FMT_MOF.1	None	-
FMT_MSA.1	None	-
FMT_MSA_EXP.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 member of authorized administrators performing these functions.
FMT_SMR.1	Modifications to the group of users that are part of a role.	Identity of the member of the authorized administrators modifying the role definition
FPT_SEP_EXP.1	None	-
FPT_TRC_EXP.1	-[5]	-
FTA_MCS.1	Rejection of a new session based on the limitation of multiple concurrent sessions.	-
FTA_TAH_EXP.1	None	-

[5] There is no audit event related to FPT_TRC_EXP.1 as this SFR is trivially fulfilled because the TOE does not comprise physically separated parts. See also chapter 8.8 for further information.

FTA_TSE.1	Denial of a session establishment due to the session establishment mechanism.	Identity of the individual attempting to establish a session

Table 8 – Auditable Events for additional SFRs

Security Functional Requirement	Auditable Event(s)	Additional Audit Record Content
FAU_STG_EXP.4	Every modifications to the setting	-
FIA_UAU.2	Every use of the authentication mechanism.	-
FIA_UAU.5	The final decision on authentication;	-
FIA_UID.2	Every use of the authentication mechanism.	-

User and/or group identity association (FAU_GEN_EXP.2)

FAU_GEN_EXP.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.

Selective audit (FAU_SEL.1-NIAP-0407)

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) [*none*];

e) [success of auditable security events;

f) failure of auditable security events; and

g) [*no additional criteria*].]

Administrable Prevention of audit data loss (FAU_STG_EXP.4)

FAU_STG_EXP.4.1 The TSF shall take one of the following actions: [

- Overwrite the oldest stored audit records
- Stop the TOE]

As specified by the administrator and [no other action] if the audit trail is full.

5.1.2 Class FDP: User Data Protection

Subset access control (FDP_ACC.1)

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].

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

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

• [the 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 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 denied to [any] group of which the authorized user is a member, deny access

c) If the requested mode of access is permitted to that authorized user, permit 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: [

• Authorized administrators, the owner of an object and owners of parent objects have access

• in case of Ownership-Chaining access is always granted].

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*].

***Full* residual information protection (FDP_RIP.*2*)**

Please note that [PP] contained only FDP_RIP.1 while this ST contains FDP_RIP.2.

FDP_RIP.2.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon the [*allocation of the resource to*] all objects.

5.1.3 Class FIA: Identification and authentication

User attribute definition (FIA_ATD.1)

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

• [login-type (SQL-Server login or Windows Account Name)

• For SQL-Server login: Hashed password]].

User authentication before any action (FIA_UAU.2)

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

Multiple authentication mechanisms (FIA_UAU.5)

FIA_UAU.5.1 The TSF shall provide [

- SQL Server Authentication and
- Access to Windows Authentication[6]

to support user authentication.

FIA_UAU.5.2 The TSF shall authenticate any user's claimed identity according to the [following rules:

- If the login is associated with a Windows user or a Windows group Windows Authentication is used,

[6] Windows Authentication is not provided by the TOE but by the environment. For this case the TOE reuses the authentication results of Windows. However, in every case the TOE enforces the policy that each user has to be successfully authenticated before allowed to perform any other action and provides an interface to the operating system to gain the authentication results and to the user to allow the user to start the process of authentication.

- If the login is a SQL Server login the SQL Server authentication is used.

].

User identification before any action (FIA_UID.2)

FIA_UID.2.1 The TSF shall require each user to identify itself before allowing any other TSF-mediated actions on behalf of that user.

5.1.4 Class FMT: Security Management

Management of security functions behaviour (FMT_MOF.1)

FMT_MOF.1.1 The TSF shall restrict the ability to *disable and enable* the functions [relating to the specification of events to be audited] to [authorized administrators].

Management of security attributes (FMT_MSA.1)

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].

Static attribute initialization (FMT_MSA_EXP.3)

FMT_MSA_EXP.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.

Management of TSF data (FMT_MTD.1)

FMT_MTD.1.1 The TSF shall restrict the ability to [*include or exclude*] the [auditable events] to [authorized administrators].

Revocation (FMT_REV.1(1))

FMT_REV.1.1(1) The TSF shall restrict the ability to revoke security attributes associated with the users within the TSC to [the authorized administrators].

FMT_REV.1.2(1) The TSF shall enforce the rules [Changes to logins are applied at the latest as soon as a new session for the login is established[7]]

Revocation (FMT_REV.1(2))

FMT_REV.1.1(2) The TSF shall restrict the ability to revoke security attributes associated with the objects within the TSC to [the authorized

[7] Please refer to chapter 7.3 of the SQL Server guidance addendum for more details

administrators and database users as allowed by the Discretionary Access Control policy].

FMT_REV.1.2(2) The TSF shall enforce the rules [The changes have to be applied immediately].

Specification of Management Functions (FMT_SMF.1)

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

- Add and delete logins
- Add and delete users
- Change role membership for DB scoped roles and Server scoped roles
- Create and destroy database scoped groups
- Create, Start and Stop Audit
- Include and Exclude Auditable events
- Define the mode of authentication
- Manage Attributes for Session Establishment
- Define the action to take in case the audit file is full]

Security roles (FMT_SMR.1)

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

- [sysadmin]; **and**
- [roles as defined in the following tables
- Roles to be defined by authorized administrators].

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

Table 9 – Default Server Roles

Role	Granted Permission(s)
bulkadmin	ADMINISTER BULK OPERATIONS
dbcreator	CREATE DATABASE
diskadmin	ALTER RESOURCES
processadmin	ALTER ANY CONNECTION, ALTER SERVER STATE
securityadmin	ALTER ANY LOGIN
serveradmin	ALTER ANY ENDPOINT, ALTER RESOURCES, ALTER SERVER STATE, ALTER SETTINGS, SHUTDOWN, VIEW SERVER STATE
setupadmin	ALTER ANY LINKED SERVER

sysadmin	CONTROL SERVER (Granted with grant option)

Table 10 - Default Database Roles

Role	Granted Permission(s)	Denied Permission(s)
db_accessadmin	ALTER ANY USER, CREATE SCHEMA CONNECT (Granted with grant option)	-
db_backupoperator	BACKUP DATABASE, BACKUP LOG, CHECKPOINT	-
db_datareader	SELECT	-
db_datawriter	DELETE, INSERT, UPDATE	-
db_ddladmin	ALTER ANY ASSEMBLY, ALTER ANY ASYMMETRIC KEY, ALTER ANY CERTIFICATE, ALTER ANY CONTRACT, ALTER ANY DATABASE DDL TRIGGER, ALTER ANY DATABASE EVENT NOTIFICATION, ALTER ANY DATASPACE, ALTER ANY FULLTEXT CATALOG, ALTER ANY MESSAGE TYPE, ALTER ANY REMOTE SERVICE BINDING, ALTER ANY ROUTE, ALTER ANY SCHEMA, ALTER ANY SERVICE, ALTER ANY SYMMETRIC KEY, CHECKPOINT, CREATE AGGREGATE, CREATE DEFAULT, CREATE FUNCTION, CREATE PROCEDURE, CREATE QUEUE, CREATE RULE, CREATE SYNONYM, CREATE TABLE, CREATE TYPE, CREATE VIEW, CREATE XML SCHEMA COLLECTION, REFERENCES	-
db_denydatareader	-	SELECT
db_denydatawriter	-	DELETE, INSERT, UPDATE
db_owner	CONTROL (Granted with grant option)	-
db_securityadmin	ALTER ANY APPLICATION ROLE, ALTER ANY ROLE, CREATE SCHEMA, VIEW DEFINITION	-

5.1.5 Class FPT: Protection of the TSF

TSF domain separation (FPT_SEP_EXP.1)

FPT_SEP_EXP.1.1	The TSF shall maintain a security domain that protects it from interference and tampering by untrusted subjects initiating actions through its own TSFI.
FPT_SEP_EXP.1.2	The TSF shall enforce separation between the security domains of subjects in the TOE Scope of Control.

Internal TSF consistency (FPT_TRC_EXP.1)

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

5.1.6 Class FTA: TOE access

Basic limitation on multiple concurrent sessions (FTA_MCS.1)

FTA_MCS.1.1	The TSF shall restrict the maximum number of concurrent sessions that belong to the same user.
FTA_MCS.1.2	The TSF shall enforce, by default, a limit of [*5*] sessions per user.

TOE access history (FTA_TAH_EXP.1)

FTA_TAH_EXP.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_EXP.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.

TOE session establishment (FTA_TSE.1)

FTA_TSE.1.1	**Refinement**: The TSF shall be able to deny session establishment based on [attributes that can be set explicitly by authorized administrators, including user identity and/or group identity, time of day, day of the week], **and [no additional attributes]**.

5.2 Security Requirements for the IT Environment

This section contains the security functional requirements for the IT environment. With the TOE being a software-only TOE, the IT environment must provide protection of the TOE from tampering and interference. These requirements can also be satisfied by the TOE since the TOE is part of the IT environment.

The requirement in this chapter has been directly taken from [PP] without any change.

Table 11 - IT Environment Security Functional Requirements

IT Environment Security Functional Requirements	
FIT_PPC_EXP.1	IT Environment Protection Profile Compliance

5.2.1 IT Environment (FIT)

IT Environment Protection Profile Compliance (FIT_PPC_EXP.1)

FIT_PPC_EXP.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].

5.3 Security Requirements for the Non-IT Environment

R.EVL The evaluation of the Operating System in the environment has to be performed to at least EAL 4 augmented by ALC_FLR.2 to provide a suitable environment that meets the requirements of the TOE described in this ST.

R.COMM Any communication path from and to the TOE will be appropriately secured to avoid eavesdropping and manipulation. This can be achieved by the use of another IT-product in the environment or by physical protection of the communication path.

5.4 TOE Security Assurance Requirements

The assurance requirements for the TOE comprise all assurance requirements for EAL 4 as defined in [CC_PART3] augmented by ALC_FLR.2.

6 TOE Summary Specification

This chapter presents an overview of the security functions implemented by the TOE and the Assurance Measures applied to ensure their correct implementation.

6.1 TOE Security Functions

This chapter presents the security functions performed by the TOE to satisfy the identified SFRs in chapter 5.1.1.

The following table gives an overview of these functions and – where appropriate – the SOF claim for the function:

Table 12 – Summary of Security Functions

Security Function	Description	SOF-Claim
Security Management	This Security Function provides the necessary functions to change the behavior of the TSF.	Not appropriate
Access Control	This Security Function realizes the Discretionary Access Control Policy for all objects under the control of the TOE.	Not appropriate
Identification and Authentication	This Security Function realizes the identification and authentication function of the TOE which is used for the cases where the identity of the user has not been verified by the environment.	SOF-medium The rating is related to the password for the SQL Server authentication.
Security Audit	This Security Function realizes the audit functionality for the TOE.	Not appropriate
Session Handling	This Security Function realizes the Session Handling.	Not appropriate

The following paragraphs contain a more detailed description of the security functions.

6.1.1 Security Management (SF.SM)

This Security Function of the TOE allows modifying the TSF data of the TOE and therewith managing the behavior of the TSF.

This comprises the following management functions:

- Add and delete logins on an instance level
- Add and delete users on a database level
- Change role membership for DB scoped roles and Server scoped roles
- Create and destroy database roles
- Create, Start and Stop Security Audit
- Include and exclude Auditable events
- Define the mode of authentication for every login
- Manage Attributes for Session Establishment
- Define the action to take in case the audit file is full

All these management functions are available via T-SQL statements directly or realized by Stored Procedures within the TOE which can be called using T-SQL. This Security Function additionally ensures that the management functions are only available for authorized administrators.

The TOE maintains a set of roles[8] on the server level and on the database level as listed in Table 9 – Default Server Roles and Table 10 - Default Database Roles. The TOE maintains a security ID for each login on a server level and each database user. This security ID is used to associate each user with his assigned roles.

6.1.2 Access Control (SF.AC)

The TOE provides a Discretionary Access Control (DAC) mechanism to control the access of users to objects based on the identity of the user requesting access, the membership of this user to roles, the requested operation and the ID of the requested object.

The TOE maintains two kinds of user representations:

1. On an instance level an end user is represented by a login. On this level the Security Function controls the access of logins to objects pertaining to the instance (e.g. to view a database)
2. On a database level an end user is represented by a database user. On this level this Security Function controls the access of database users to objects of the database (e.g. to read or create a table).

Members of the database roles “db_owner” or “db_accessadmin” are able to add users to a database. The TOE maintains an internal security identifier (SID) for every user and role. Each database user can be associated with one instance “login”.

[8] Please note that in [PP] the terms “group” and "role" are used while the TOE refers only to “role”. However the concept of "roles" in the TOE covers all aspects as required for "groups" and "roles" in [PP]

Every object controlled by the TOE has an ID, an owner and a name.

Objects in the TOE form a hierarchy and belong to one of three different levels: server, database and schema.

The TOE maintains an Access Control List (ACL) for each object within its scope. These ACLs are stored in a system table which exists in every database for database related ACLs and in a system table in the ‘master’ database for instance level ACLs.

Each entry of an ACL contains a user SID and defines whether a permission is an “Allow” or a “Deny” permission for that SID.

When a new object is created, the creating user is assigned as the owner of the object and has complete control over the object. The ACL for a newly created object is always empty by default.

After creation, grant, deny or revoke permissions on objects can be assigned to users. Changes to the security relevant attributes of objects are immediately applied.

When a user attempts to perform an action to an object under the control of the TOE, the TOE decides whether the action is to be permitted based on the following rules:

1. If the requested mode of access is denied to that authorized user, the TOE will deny access
2. If the requested mode of access is denied to any role of which the authorized user is a member, the TOE will deny access
3. If the requested mode of access is permitted to that authorized user, the TOE will permit access
4. If the requested mode of access is permitted to any role of which the authorized user is a member, the TOE will permit access
5. Else: The TOE will deny access

The TOE permission check for an action on an object includes the permissions of its parent objects. The permissions for the object itself and all its parent objects are accumulated together before the aforementioned rules are evaluated. Note: Some actions require more than one permission.

This means that if a user or a role has been granted a permission to an object this permission is also valid for all child objects. E.g. if a user has been granted a permission to a schema, he automatically has the same permission on all tables within that schema, if the permission has not explicitly been denied. Similarly, if a user has been denied a permission on a schema, he will be denied the same permission to all tables within that schema, regardless of explicit grant permissions.

The rules as described before are always applied when a user requests access to a certain object using a certain operation. There are only two situations where these access control rules are overridden:

1. The system administrator, the owner of an object and owners of parent objects always have access, so for these users the TOE will always allow access to the object

2. In the case of “Ownership Chaining” which is described in chapter 9.2 in more detail the access is allowed.

6.1.3 Identification and Authentication (SF.I&A)

This Security Function requires each user to be successfully authenticated before allowing any other actions on behalf of that user. This is done on an instance level and means that the user has to be associated with a login of the TOE.

The TOE knows two types of logins: Windows accounts and SQL Server logins. The administrator has to specify the type of login for every login he is creating.

The possibility for the TOE to perform its own authentication is necessary because not all users connecting to the TOE are connecting from a Windows environment.

Microsoft Windows account names

These logins are associated with a user account of the Windows Operating System in the environment.

For these logins the TOE requires that the Windows environment passes on the Windows SID(s) of that user to authenticate the user before any other action on behalf of that user is allowed.[9]

For these logins the Windows security identifier (SID) from the Windows account or group is used for identification of that login within the TOE. Any permission is associated with that SID.

Any changes which occur to a Windows account in the environment while a user is connected to the TOE are not applied by the TOE until the user logs off and logs on again.

SQL Server login names

SQL Server logins are not associated with a user of Windows but are maintained by the TOE itself. For every SQL Server login the TOE maintains a login name and a password. The password is not stored in plain text, but hashed using the SHA-1 hash function provided by the Operating System in the environment.

Each SQL Server login name is stored in a system table. SQL Server generates a SID that is used as a security identifier and stores it in this table.

This SID is internally used as a security identifier for the login.

If a user is connecting to the TOE using a SQL Server login he has to provide the username and password. The TOE hashes the password using the hash function provided by the Operating System in the environment, and compares the hash to the value stored for that user. If the values are identical the TOE has successfully authenticated the user.

[9] Windows authenticates users based on a username and password. After successful authentication of a user Windows associates a list of SID(s) with every user which represent the user and every group the user is a member of. Details can be found in [WIN_ST].

Any changes that occur to the definition of SQL Server login are immediately applied by the TOE.

6.1.4 Security Audit (SF.AU)

The TOE produces audit logs for all security relevant actions. These audit logs are stored into files in the environment of the TOE.

The Security Audit of the TOE especially comprises the following events:

- Startup and Shutdown of the TOE
- Start and Shutdown of Security Audit Function
- Every login attempt including the processes for authentication and session establishment
- Every successful request to perform an operation on an object, covered by the access control function
- Modifications to the role membership of users
- The use of the Security Function SF.SM
- Every rejected attempt to establish a session

The TOE maintains a set of events which can be additionally audited and provides the administrator with the capability to start a Security Audit process to capture these events.

For each event in the Security Audit logs the following information is stored:

1. Date and Time of the event
2. Type of Event
3. Identity of the user causing the event (if available)
4. ID of the object
5. Outcome (success or failure) of the event
6. For the rejection of a session additionally the reason for the rejection

The TOE maintains a category for every audit event.

The administrator has the possibility to specify, what should happen in case an audit file is full. The following two scenarios are supported in the evaluated version:

1. Rollover

The administrator specifies a maximum size per trace file and a maximum number of files for the Security Audit. If one audit file is full, the TOE starts the next file until the maximum number of files has been reached. When the maximum number of files has been reached and the last audit file is full, the TOE will start overwriting the oldest audit file.

2. Shutdown

The administrator specifies one trace file with a maximum size and the option to shut down the TOE on any audit error. When the maximum size of the trace file has been reached the TOE will stop operation.

The TOE provides the possibility to create a filter for the audit function. Using this filter mechanism the administrator is able to exclude auditable events from being audited based on the following attributes:

- User identity
- Event type,
- Object identity,
- Success or failure of auditable security events

However to modify the behavior of the Security Audit function by including additional or excluding events from being audited the administrator has to stop the Security Audit process, modify the Security Audit function and start the Security Audit process again.

6.1.5 Session Handling (SF.SE)

After a user attempting to establish a session has been successfully authenticated by SF.I&A this Security Function decides whether this user is actually allowed to establish a session to the TOE.

The TOE uses two sets of additional criteria to decide whether a user is allowed to establish a session. First the TOE enforces a limit of the number of concurrent sessions a user is allowed to have at one time. This limit is set to 5 by default but can be modified by authorized administrators as described in SF.SM. If a user reached the limit of concurrent sessions the TOE will deny establishing another session for that user.

Furthermore the admin is able to specify a set of rules to explicitly deny session establishment based on:

- User’s identity,
- Time of the day and
- Day of the week.

The TOE only establishes a session for a user if no explicit deny rule for that user has been specified.

For every attempt to establish a session (whether successful or not) the TOE stores the date and time of the event and the number of unsuccessful attempts since the last successful attempt.

After the TOE established a session to a user the user context is held in a context with limited permission. SF.SE maintains a separate context for the execution of each operation by a user. As soon as a user performs an operation on an object the TOE starts at least one thread to perform this operation.

When the TOE reuses memory which could contain previous information content, this previous information will not be available for any user. To ensure this, the TOE either directly overwrites the memory completely with new information or with a certain pattern. Before the

previous information has been overwritten the resource is not available for any usage. For memory which is allocated using the Operating System the TOE uses a function of the OS, which ensures that only empty memory is provided to the TOE. Whenever data is written to or loaded from disc this is done pagewise where a page has the size of 8 KB.

6.2 Assurance Measures

For the evaluation of the TOE the assurance requirements according to CC EAL4 augmented with ALC_FLR.2 apply. This chapter identifies the assurance measures that are or will be applied by Microsoft in the course of the evaluation to satisfy the assurance requirements. The corresponding assurance measures are listed in Table 13 below (N.B. Some of the documentation listed therein is not prepared yet, therefore currently corresponding document titles and versions are not available).

Table 13 - Assurance Measures

SAR(s)	Assurance Measure(s)
ACM_AUT.1 ACM_CAP.4 ACM_SCP.2	Provision of CM system documentation
ADO_DEL.2	Provision of delivery documentation
ADO_IGS.1	Provision of installation, generation and startup documentation (either as part of administrator guidance documentation or as a separate document)
ADV_FSP.2	Provision of functional specification documentation
ADV_HLD.2	Provision of high-level design documentation
ADV_IMP.1	Provision of a subset of the implementation of the TOE
ADV_LLD.1	Provision of low-level design documentation
ADV_RCR.1	Provision of representation of correspondence documentation
ADV_SPM.1	Provision of an informal security policy model documentation
AGD_ADM.1 AGD_USR.1	Provision of user/administrator guidance documentation
ALC_DVS.1	Provision of development security documentation
ALC_FLR.2	Provision of flaw remediation documentation
ALC_LCD.1	Provision of life-cycle model documentation
ALC_TAT.1	Provision of tool and techniques documentation
ATE_COV.2 ATE_DPT.1 ATE_FUN.1	Provision of test documentation
ATE_IND.2	Provision of the TOE and its platform, Provision of test tools, scripts, etc.,
AVA_MSU.2	Provision of misuse analysis documentation,
AVA_SOF.1	Provision of a SOF analysis
AVA_VLA.2	Provision of vulnerability analysis documentation

7 Protection Profile (PP) Claims

This Security Target is compliant to the U.S. Government Protection Profile for Database Management Systems in Basic Robustness Environments, Version 1.1, 07.06.2006 ([PP]).

8 Rationale

This chapter demonstrates the completeness and consistency of this ST by providing justification for the following:

Traceability The security objectives for the TOE and its environment are explained in terms of threats countered and assumptions met. The SFRs are explained in terms of objectives met by the requirement. The traceability is illustrated through matrices that map the following:

- security objectives to threats encountered
- environmental objectives to assumptions met
- SFRs to objectives met
- Security functions to SFRs met

Assurance Level A justification is provided for selecting an EAL4+ level of assurance for this ST.

SOF A rationale about the SOF claim is provided.

Dependencies A mapping is provided as evidence that all dependencies are met.

8.1 Rationale for TOE Security Objectives

The following table summarizes the rationale for the security objectives.

Threats, Assumptions, OSP / Security Objectives	O.ACCESS_HISTORY	O.ADMIN_GUIDANCE	O.ADMIN_ROLE	O.AUDIT_GENERATION	O.CONFIGURATION_IDENTIFICATION	O.DOCUMENTED_DESIGN	O.INTERNAL_TOE_DOMAINS	O.MANAGE	O.MEDIATE	O.PARTIAL_FUNCTIONAL_TEST	O.PARTIAL_SELF_PROTECTION	O.RESIDUA_INFORMATION	O.TOE_ACCESS	O.VULNERABILITY_ANALYSIS	O.I&A	OE.NO_EVIL	OE.NO_GENERAL_ PURPOSE	OE.OS_PP_VALIDATED	OE.PHYSICAL	OE.COMM
T.ACCIDENTAL_ADMIN_ERROR		X																		
T.MASQUERADE													X		X					
T.POOR_DESIGN					X	X								X						
T.POOR_IMPLEMENTATION					X					X				X						
T.POOR_TEST		X				X				X				X						
T.RESIDUAL_DATA												X						X		
T.TSF_COMPROMISE							X	X			X	X								
T.UNAUTHORIZED_ACCESS	X								X						X					
T.UNIDENTIFIED_ACTIONS		X						X												
P.ACCOUNTABILITY				X									X		X					
P.ROLES			X																	
A.NO_EVIL																X				
A.NO_GENERAL_PURPOSE																	X			
A.OS_PP_VALIDATED																		X		
A.PHYSICAL																			X	
A.COMM																				X

Table 14 – Summary of Security Objectives Rationale

Details are given in the following table. These details are directly taken from [PP] with only the changes indicated by bold italic text.

Table 15 – Rationale for TOE Security Objectives

Threat/Policy	**Objectives Addressing the Threat/Policy**	**Rationale**
T.ACCIDENTAL_ADMIN_ERROR An administrator may incorrectly install or configure the TOE resulting in ineffective security mechanisms.	O.ADMIN_GUIDANCE The TOE will provide administrators with the necessary information for secure management.	O.ADMIN_GUIDANCE helps to mitigate this threat by ensuring the TOE administrators have guidance that instructs them how to administer the TOE in a secure manner. Having this guidance helps to reduce the mistakes that an administrator might make that could cause the TOE to be configured in insecurely.
T.MASQUERADE A user or process may masquerade as another entity in order to gain unauthorized access to data or TOE resources	O.TOE_ACCESS The TOE will provide mechanisms that control a user's logical access to the TOE	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. ***[Part of the rationale deleted]***
	O.I&A ***The TOE will provide a mechanism for identification and authentication of users.***	***O.I&A mitigated this threat by providing the means to identify and authenticate the user where the I&A mechanisms of the environment is not used. The correct identity of the user is the basis for any decision of the TOE about an attempt of a user to access data.***
T.POOR_DESIGN ***Errors*** in requirements specification or design of the TOE may occur, leading to flaws that may be exploited by ***an attacker.***	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 flaws which can be exploited in the intended environment for the TOE and that the TOE is resistant to penetration attacks performed by attackers possessing a low attack potential.	O.VULNERABILITY_ANALYSIS ensures that the design of the TOE is analyzed for design flaws.
T.POOR_IMPLEMENTATIO N ***Errors*** in implementation of the TOE design may occur, leading to flaws that may be exploited by ***an attacker.***	O.CONFIGURATION_IDENTIFICATI ON The configuration of the TOE is fully identified in a manner that will allow implementation errors to be identified and corrected with the TOE being redistributed promptly.	O.CONFIGURATION_IDENTIFICAT ION 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.
	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_T EST increases the likelihood that any errors that do exist in the implementation (with respect to the functional specification ***and high-level design***) will be discovered through testing.
	O.VULNERABILITY_ANALYSI S The TOE will undergo some vulnerability analysis to demonstrate the design and implementation of the TOE does not contain any flaws which can be exploited in the intended environment for the TOE and that the TOE is resistant to penetration attacks performed by attackers possessing a low attack potential.	O.VULNERABILITY_ANALY SIS helps reduce errors in the implementation that may not be discovered during functional testing. Ambiguous design documentation and the fact that exhaustive testing of the external interfaces is not required may leave bugs in the implementation undiscovered in functional testing.
T.POOR_TEST Lack of or insufficient tests to demonstrate that all TOE security functions operate	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

correctly (including in a fielded TOE) may result in incorrect TOE behaviour being discovered thereby causing potential security vulnerabilities.		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_T EST increases the likelihood that any errors that do exist in the implementation (with respect to the functional specification and ***high level design***) will be discovered through testing.
	O.VULNERABILITY_ANALYSI S The TOE will undergo some vulnerability analysis to demonstrate the design and implementation of the TOE does not contain any flaws which can be exploited in the intended environment for the TOE and that the TOE is resistant to penetration attacks performed by attackers possessing a low attack potential.	O.VULNERABILITY_ANALSIS addresses this concern by requiring a vulnerability analysis be performed in conjunction with testing that goes beyond functional testing. This objective provides a measure of confidence that the TOE does not contain security flaws that may not be identified through functional testing. While these testing activities are a necessary activity for successful completion of an evaluation, this testing activity does not address the concern that the TOE continues to operate correctly and enforce its security policies once it has been fielded. Some level of testing must be available to end users to ensure the TOE's security mechanisms continue to operator correctly once the TOE is fielded.
	O.ADMIN_GUIDANCE ***The TOE will provide administrators with the necessary information for secure management.***	***O.ADMIN_GUIDANCE*** ***Addresses this threat as it provides the administrator with guidance, how to test the correct operation of the security functions of a fielded TOE.***

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.***
	OE.OS_PP_VALIDATED	***For memory which is allocated from the OS to the TOE the OS ensures that this memory does not contain any residual information. (see also Table*** 23***).*** ***Due to the fact that the threat T.RESIDUAL_DATA refers to reallocation of a resource, the combination of this objective for the environment and O.RESIDUAL_INFORMATION is sufficient to counter the threat.***
T.TSF_COMPROMISE A malicious user or process may cause configuration data 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_PROTECTI ON 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 behaviour of TSF functions.
	O.INTERNAL_TOE_DOMAINS The TSF will maintain internal	O.INTERNAL_TOE_DOMAINS ensures the TOE will establish

	domains for separation of data and queries belonging to concurrent users.	separate domains for data belonging to users.
T.UNAUTHORIZED_ACCSS 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. The TOE requires successful authentication to the TOE prior to gaining access to any controlled-access content 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.***
	O.I&A ***The TOE will provide a mechanism for identification and authentication of users.***	***O.I&A*** ***contributes to countering this threat by providing the means to identify and authenticate the user where the I&A mechanisms of the environment is not used. The correct identity of the user is the basis for any decision of the TOE about an attempt of a user to access data.***
T.UNIDENTIFIED_ACTIONS Failure of the authorized administrator to identify and act upon unauthorized actions may	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

occur.		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 that only authorized administrators shall be able to manage the Security Audit functions. In this way an attacker is not able to stop the Security Audit function or to exclude his events from being audited.***
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 authorized administrators with the capability of configuring the Security Audit mechanism to record the actions of a specific 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 Security Audit mechanism etc.).***
	O.I&A ***The TOE will provide a mechanism for identification and authentication of users.***	***O.I&A supports this policy by providing the means to identify and authenticate the user where the I&A mechanisms of the environment cannot be used. The identity of the user is stored in the audit logs.***
	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 ensure that any user has been successfully identified and authenticated prior to allowing any TOE access or any TOE mediated access on behalf of those users.***
P.ROLES The TOE shall provide an authorized administrator role for secure administration of the TOE. This role shall be	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

separate and distinct from other authorized users.		the role of authorized administrator is required (O.ADMIN_ROLE).

8.2 Rationale for the Security Objectives for the Environment

The following table contains the rationale for the IT Environmental Objectives. This rationale has directly been taken from [PP] with only the changes indicated by bold italic text.

Table 16 – Rationale for IT Environmental Objectives

Assumption	**Environmental Objective Addressing the Assumption**	**Rationale**
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.
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.
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. ***The evaluation and certification of the underlying OS has been done on at least EAL 4 augmented by ALC_FLR.2.***	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-bypassibility, Audit Review, Audit Storage, and Identification and Authentication. ***The evaluation and certification of the underlying OS has to be done on at least EAL 4 augmented by ALC_FLR.2. This means that the Operating System in the environment has been evaluated on at least the same level as the TOE described in this ST is***

		evaluated on.
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.
A.COMM ***It is assumed that any communication path from and to the TOE is appropriately secured to avoid eavesdropping and manipulation.***	***OE.COMM*** ***Any communication path from and to the TOE will be appropriately secured to avoid eavesdropping and manipulation.***	***A.COMM is completely and directly addressed by OE.COMM. OE.COMM and A.COMM both address the requirement that any communication path to and from the TOE has to be appropriately secured.***

8.3 Rationale for the TOE and environmental Security Requirements

The following table contains the rationale for the TOE Security Requirements. This rationale has been directly taken from [PP] with only the changes indicated by bold italic text.

Table 17 – Rationale for TOE Security Requirements

Objective	**Requirements Addressing the Objective**	**Rationale**
O.ACCESS_HISTORY The TOE will store and retrieve information (to authorized users) related to previous attempts to establish a session.	FTA_TAH_EXP.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_EXP.1)
O.ADMIN_GUIDANCE The TOE will provide administrators with the necessary information for secure management.	***ADO_DEL.2***	***ADO_DEL.2*** ensures that the administrator is provided documentation that instructs them how to ensure the delivery of the TOE, in whole or in parts, has not been tampered with or corrupted during delivery. This requirement ensures the administrator has the ability to begin their TOE installation with a clean (e.g., malicious code has not been inserted once it has left the developer's control) version of the TOE, which is necessary for secure management of the TOE.

	ADO_IGS.1	ADO_IGS.1 ensures the administrator has the information necessary to install the TOE in the evaluated configuration. Often times a vendor's product contains software that is not part of the TOE and has not been evaluated. The Installation, Generation, and Startup (IGS) documentation ensures that once the administrator has followed the installation and configuration guidance the result is a TOE in a secure configuration.
	AGD_ADM.1	AGD_ADM.1 mandates the developer provide the administrator with guidance on how to operate the TOE in a secure manner. This includes describing the interfaces the administrator uses in managing the TOE, security parameters that are configurable by the administrator, how to configure the TOE's rule set and the implications of any dependencies of individual rules. The documentation also provides a description of how to setup and review the auditing features of the TOE. 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_USR.1	AGD_USR.1 is 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).

	AVA_MSU.2	***AVA_MSU.2*** ensures that the guidance documentation is complete and consistent, and notes all requirements for external security measures.
O.ADMIN_ROLE The TOE will provide authorized administrators 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_EXP.2	FAU_GEN_EXP.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.
	FAU_STG_EXP.4	***FAU_STG_EXP.4 allows the administrator to define what should happen in the case where the audit file is full. This provides the administrator with the possibility to decide about possible audit data loss or stopping of services based on the information stored in the database.***
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.	***ACM_CAP.4***	***ACM_CAP.4*** 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 adequately and accurately documented.	***ADV_FSP.2***	***ADV_FSP.2*** requires that the interfaces to the TOE be documented and specified.
	ADV_HLD.2	***ADV_HLD.2*** requires the high level design of the TOE be documented and specified and that said design be shown to correspond to the interfaces.
	ADV_LLD.1	***The low-level design of a TOE provides the description of the internal workings of the TSF in terms of modules and their interrelationships and***

		dependencies. The low-level design provides assurance that the TSF subsystems have been correctly and effectively refined and therewith contributes to the objective that the design of the TOE has to be adequately and accurately documented.
	ADV_IMP.1	***ADV_IMP.1.1D requires that the developer provide the implementation representation for a subset of the TSF. The intention is that access to at least a portion of the TSF will provide the evaluator with an opportunity to examine the implementation representation for those portions of the TOE where such an examination can add significantly to the understanding of, and assurance in, the mechanisms employed. In this way this assurance requirement contributes to the objective that the design of the TOE has to be adequately and accurately documented.***
	ADV_RCR.1	ADV_RCR.1 requires that there be a correspondence between adjacent layers of the design decomposition.
O.MANAGE The TOE will provide all the functions and facilities necessary to support the authorized administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.	FMT_MOF.1	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_EXP.3	FMT_MSA_EXP.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.
	FPT_TRC_EXP.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.INTERNAL_TOE_DOMAINS The TSF will maintain internal domains for separation of data and queries belonging to concurrent users.	FPT_SEP_EXP.1	FPT_SEP_EXP.1 requires the TOE to maintain a separate domain for its own execution separate from other processes.
O.PARTIAL_FUNCTIONAL_TEST The TOE will undergo some security functional testing that demonstrates the TSF satisfies some of its security	***ATE_COV.2***	***ATE_COV.2*** requires that there be a correspondence between the tests in the test documentation and the TSF as described in the functional specification.

functional requirements.	***ATE_DPT.1***	***ATE_DPT.1 requires that there be a correspondence between the tests in the test documentation and the TSF as described in the high level design.***
	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_PROTECTI ON 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.	FPT_SEP_EXP.1	The explicitly specific component FPT_SEP_EXP.1 was chosen to ensure the TSF provides a domain that protects itself from untrusted users. If the TSF cannot protect itself it cannot be relied upon to enforce its security policies. The explicitly specified version was used to distinguish the aspects of FPT_SEP provided by the TOE versus the aspects provided by the IT environment.
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.2	FDP_RIP.2 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	FIA_ATD.1	FIA_ATD.1 defines the attributes of users, including a user ID that is used by the TOE to determine

TOE.		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 to the TOE based on certain criteria.
	AVA_SOF.1	***AVA_SOF.1 requirement is applied to the password mechanism used by the local administrator (The single use authentication mechanism supplies by the IT environment (i.e., authentication server) has this same assurance requirement levied against it to ensure a consistent level of assurance.) For this TOE, the strength of function specified is medium. This requirement ensures the developer has performed an analysis of the password mechanism to ensure the probability of guessing a local administrator's password meets the requirements for an SOF claim of SOF-medium according to the rating as defined in Annex B of the CEM.***
O.VULNERABILITY_ANALYSI S The TOE will undergo some vulnerability analysis to demonstrate the design and implementation of the TOE does not contain any flaws which can be exploited in the intended environment for the TOE and that the TOE is resistant to penetration attacks performed by attackers possessing a low attack potential.	***AVA_VLA.2***	***The AVA_VLA.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_VLA.2 requires the developer to perform a search for potential vulnerabilities in all the TOE deliverables. For those vulnerabilities that are not eliminated, a rationale must be provided that describes why***

		these vulnerabilities cannot be exploited by a threat agent with a low attack potential, which is in keeping with the desired assurance level of this TOE. As with the functional testing, a key element in this component is that an independent assessment of the completeness of the developer's analysis is made, and more importantly, an independent vulnerability analysis coupled with testing of the TOE is performed. This component provides the confidence that security flaws do not exist in the TOE that could be exploited by a threat agent of low (or lower) attack potential to violate the TOE's security policies.
O.I&A ***The TOE will provide a mechanism for identification and authentication of users.***	***FIA_UAU.2***	***FIA_UAU.2 realizes the authentication part of O.I&A as it requires that each user has to get successfully authenticated before allowing any other TSF-mediated action on behalf of that user.***
	FIA_UID.2	***FIA_UID.2 realizes the identification part of O.I&A as it requires that each user has to get successfully identified before allowing any other TSF-mediated action on behalf of that user.***
	FIA_UAU.5	***FIA_UAU.5 specifies that the TOE uses two methods to ensure that every user has to be successfully authenticated.*** ***On the one hand the TOE is able to reuse the authentication results from the environment and on the other hand the TOE provides a password based authentication mechanism.***

The following table includes the rationale for the IT Environment Requirements. This rationale has been directly taken from [PP] with only the changes indicated by bold italic text .

Table 18 – Rationale for 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 administrator guidance.	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.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. ***The evaluation and certification of the underlying OS has to be done on at least EAL 4 augmented by ALC_FLR.2.***	FIT_PPC_EXP.1 ***R.EVL***	FIT_PPC_EXP.1 states the underlying OS must be validated against a OS PP of at least basic robustness and ***R.EVL defines that the evaluation of the underlying OS has to done on at least EAL 4 augmented by ALC_FLR.2 Table*** 23 ***gives more details about which parts of the Operating System are involved.***
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.COMM ***Any communication path from and to the TOE will be***	***R.COMM***	***R.COMM defines that any communication path to and from the TOE will be secured either by the use of another***

appropriately secured to avoid eavesdropping and manipulation.		***IT product of by physical protection.***

8.3.1 Mutual support and internal consistency of security requirements

From the details given in this rationale it becomes evident that the functional requirements form an integrated whole and, taken together, are suited to meet all security objectives. Requirements from [CC_PART2] and extended requirements are used to fulfill the security objectives.

The core TOE functionality is represented by the requirements for Access Control (FDP_ACC.1, FDP_ACF.1-NIAP-0407), Security Audit (FAU_GEN.1-NIAP-0410, FAU_GEN_EXP.2, FAU_SEL.1-NIAP-0407 FAU_STG_EXP.4), Identification and Authentication (FIA_ATD.1, FIA_UAU.2, FIA_UAU.5, FIA_UID.2), Security Management (FMT_MOF.1, FMT_MSA.1, FMT_MSA_EXP.3, FMT_MTD.1, FMT_REV.1(1), FMT_REV.1(2), FMT_SMF.1, FMT_SMR.1 and Session Handling (FPT_SEP_EXP.1, FPT_TRC_EXP.1, FTA_MCS.1, FTA_TAH_EXP.1, FTA_TSE.1).

The ST does not contain any SFR with requirements which conflict with other SFRs.

Together with the SARs out of [CC_PART3] the SFRs are suitable to counter the threats against the TOE as shown in the rationale in Table 17.

Therefore it becomes clear that the SFRs in this ST mutually support each other and form a consistent whole.

8.4 Rationale for Assurance Requirements

The table in chapter 6.2 shows how all assurance requirements were satisfied and that there is at least one assurance measure defined in the TOE Summary Specification to meet each of the security assurance requirements.

Based on the definition of the attack potential of a potential attacker in chapter 3.3 it is necessary to perform a vulnerability analysis to show that the TOE is resistant against attacks with a low attack potential. Therefore it has been necessary to use the assurance component AVA_VLA.2.

As shown in chapter 8.2 the SAR ALC_FLR.2 is necessary for the objective O.CONFIGURATION_IDENTIFICATION.

Because of the dependencies of the component AVA_VLA.2, and to gain a higher level of assurance in the correct implementation of the Security Functions, the author decided to specify an Evaluation Assurance Level 4 augmented by ALC_FLR.2.

8.5 Rationale for Strength of Function Claim

The SOF-claim for the authentication mechanism of the TOE (which is the only function which is based on a permutational or probalistic algorithm) is **SOF-medium**.

Based on the definition of the attack potential in chapter 3.3 it would have been sufficient to claim it to be SOF-basic as the attacker has only a low attack potential. Also the definitions of SFRs in chapter 5.1 and the definition of the Security Objectives do not contain any explicit requirement regarding the SOF claim.

However the authentication function is an obvious mechanism to start a brute force attack on and the author thinks that this mechanism should be stronger than the overall resistance of the TOE against threats. For this reason the author specified the strength of this function to be **SOF-medium**.

8.6 Rationale for satisfying all Dependencies

The following table contains the rationale for satisfying all dependencies of the Security Functional Requirements. This rationale has been taken from [PP] with only the changes indicated by bold italic text.

Table 19 – Functional Requirements Dependencies

Requirement	Dependency	Satisfied
FAU_GEN.1-NIAP-0410	FPT_STM.1	This requirement must be satisfied by the IT environment because the DBMS is a software only TOE.
FAU_GEN_EXP.2	FAU_GEN.1-NIAP-0410 FIA_UID.1	Satisfied (FIA_UID.2 is hierarchical to FIA_UID.1) The dependency to FIA_UID.1 is either fulfilled by the TOE (for SQL logins) and by the environment (For Windows logins).
FAU_SEL.1-NIAP-0407	FAU_GEN.1-NIAP-0410 FMT_MTD.1	Satisfied
FAU_STG_EXP.4	***FAU_STG.1***	***The dependency to FAU_STG.1 is satisfied by the environment. The TOE as a DBMS has to rely on the Operating System to protect the files.***
FDP_ACC.1	FDP_ACF.1-NIAP-0407	Satisfied.
FDP_ACF.1-NIAP-0407	FDP_ACC.1 FMT_MSA.3	The dependency on FMT_MSA.3 is satisfied by FMT_MSA_EXP.3.
FDP_RIP.2	***None***	***N/A***
FIA_ATD.1	None	N/A
FIA_UAU.2	***FIA_UID.1***	***Satisfied, see explanation for FAU_GEN_EXP.2*** ***The dependency to FIA_UID.1 is either fulfilled by the TOE or by the environment.***

FIA_UAU.5	***None***	***N/A***
FIA_UID.2	***None***	***N/A***
FMT_MOF.1	FMT_SMF.1 FMT_SMR.1	Satisfied.
FMT_MSA.1	[FDP_ACC.1 or FDP_IFC.1] FMT_SMF.1 FMT_SMR.1	***Dependency satisfied by the combination of FDP_ACC.1, FMT_SMF.1 and FMT_SMR.1***
FMT_MSA_EXP.3	FMT_MSA.1 FMT_SMR.1	Satisfied.
FMT_MTD.1	FMT_SMF.1 FMT_SMR.1	Satisfied.
FMT_REV.1(1)	FMT_SMR.1	Satisfied.
FMT_REV.1(2)	FMT_SMR.1	Satisfied.
FMT_SMF.1	None	N/A
FMT_SMR.1	FIA_UID.1	***Satisfied*** ***(FIA_UID.2 is hierarchical to FIA_UID.1)***
FPT_SEP_EXP.1	None	N/A
FPT_TRC_EXP.1	FPT_ITT.1	***This dependency does not need to be fulfilled because the TOE does not comprise physically separated parts.***
FTA_MCS.1	FIA_UID.1	***Satisfied*** ***(FIA_UID.2 is hierarchical to FIA_UID.1)*** ***The dependency to FIA_UID.1 is either fulfilled by the TOE or by the environment.fr***
FTA_TAH_EXP.1	None	N/A
FTA_TSE.1	None	N/A

Table 20 – Functional Requirements Dependencies for IT Environment

Requirement	Dependency	Satisfied
FIT_PPC_EXP.1	None	N/A

The set of assurance requirements for EAL 4 is consistent and all dependencies are met. The only additional assurance requirement ALC_FLR.2 does not have any dependency.

8.7 Rationale for Explicit Requirements

Table 21 presents the rationale for the inclusion of the explicit functional and assurance requirements. The explicit requirements that are included as NIAP interpretations do not require a rationale for their inclusion per CCEVS management. The rationale has been directly taken from [PP] with no other than the indicated changes.

Table 21 – Rationale for Explicit Requirements

Explicit Requirement	Identifier	Rationale
FAU_GEN_EXP.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 explicit requirement was created to allow the audit function to use the user identity or the group identity or both. ***However this SFR has been developed based on FAU_GEN.2 and has the same component, family and class behavior***
FPT_TRC_EXP.1	Internal TSF consistency	FPT_TRC_EXP.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. ***However this SFR has been developed based on FPT_TRC.1 and has the same component, family and class behavior***
FPT_SEP_EXP.1	TSF domain separation	Given the nature of a PP compliant TOE that is described in the TOE Description, the objectives and functional requirements must ultimately reflect this description. Software Only Toe properties are instantiated in Section ***5*** of the PP (i.e., the Functional Requirements section) by creating explicitly stated requirements in place of FPT_SEP.1. The need for explicitly stated requirements is that when invoked, the current FPT_SEP.1 Common Criteria Requirement requires the TOE (not its environment) to protect itself from external interference and tampering. Typically, “Software Only” technology cannot fully meet these requirements as written. Software Only TOEs should be expected to work in the context of their hardware environment to aid in enforcing domain separation but cannot be required to counter fully the threats without hardware. Therefore, the PP authors chose to use

		explicitly stated requirements for domain separation when attempting to accommodate the “Software Only” TOE. ***However this SFR has been developed based on FPT_SEP.1 and has the same component, family and class behavior***
FTA_TAH_EXP.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. ***However this SFR has been developed based on FTA_TAH.1 and has the same component, family and class behavior.***
FMT_MSA_EXP.3	Static attribute initialization	The CC does not allow the PP author to specify restrictive values that are not modifiable. This explicit 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. ***However this SFR has been developed based on FMT_MSA.3 and has the same component, family and class behavior.***
FAU_STG_EXP.4	***Administrable Prevention of audit data loss***	***It has been necessary to develop this explicit Security Functional Requirement because part II of [CC] does not contain any SFR which allows specifying a set of allowed actions which can be taken in the case where the audit is full.*** ***For the TOE described in this ST it was necessary to provide authorized administrators with the possibility to specify what should happen if the audit log is full. However there should only be one action to be taken in this case.*** ***However this SFR has been developed based on the definition of FAU_STG.4 and has the same family behaviour except that it is not hierarchical to any other SFR. .***

The following requirements were modified to refer to the IT environment. Throughout each requirement ‘TSF’ was replaced with ‘IT Environment’, ‘TSC’ was replaces with ‘IT Environment Scope of Control’, etc.

Table 22 – Rationale for Environmental Requirements

Environmental Requirement	Identifier	Rationale
FIT_PPC_EXP.1	IT Environment Protection Profile Compliance	***This requirement is necessary to ensure the environment of the TOE will provide the security functionality needed for the secure operation of the TOE.*** ***This requirement is fulfilled as the underlying OS for the TOE described in this ST is Windows 2003 Server SP1 which has been certified in accordance to [WIN_PP] (see [WIN_VR] for evidence).***

The exact definition and behavior of FIT_PPC_EXP.1 is described in chapter 9.1.
Based on the definition of the assumption A.OS_PP_VALIDATED and the Security Functions defined in this ST the SFRs of the [WIN_ST] listed in the following table are needed to support the Security Functions of the TOE.

Table 23 - Important SFRs of the environment

Aspect	SFR of the environment	Title
Access Control (including Access Control for audit files)	FDP_ACC.2(a)	Discretionary Access Control Policy
Authentication of users	FIA_UAU.1	Timing of authentication
Identification of users	FIA_UID.1	Timing of identification
Non Bypassabiltity	FPT_RVM.1	Non-bypassability of the TSP
Domain Separation	FPT_SEP.1	TSF domain separation
Cryptographic Functions (Hashing)	FCS_COP.1(e)	Cryptographic Operation (Server SHA Hash)
Time Stamps	FPT_STM.1	Reliable Time Stamps
Residual Information Protection	FDP_RIP.2 Note1_EX	Object/Subject Residual Information Protection

8.8 TOE Summary Specification Rationale

The following table summarizes which SFR is addressed by which Security Function:

The following paragraphs give the more detailed justification for this rationale.

Table 24 - Assignment of SFRs to Security Functions

Requirement/Security Function	SF.SM	SF.AC	SF.I&A	SF.AU	SF.SE
FAU_GEN.1-NIAP-0410				X	
FAU_GEN_EXP.2				X	
FAU_SEL.1-NIAP-0407	X			X	
FAU_STG_EXP.4.1	X			X	
FDP_ACC.1		X			
FDP_ACF.1-NIAP-0407		X			
FDP_RIP.2					X
FIA_ATD.1			X		
FIA_UAU.2			X		
FIA_UAU.5			X		
FIA_UID.2			X		
FMT_MOF.1	X				
FMT_MSA.1	X				
FMT_MSA_EXP.3		X			
FMT_MTD.1	X				
FMT_REV.1(1)	X		X		
FMT_REV.1(2)		X			
FMT_SMF.1	X				
FMT_SMR.1	X				
FPT_SEP_EXP.1			X		X
FPT_TRC_EXP.1					
FTA_MCS.1					X
FTA_TAH_EXP.1					X
FTA_TSE.1					X

The following rationale shows why this mapping is correct:

Table 25 – Rationale for TOE Summary Specification

Requirement	Fulfilled by Security Function	Rationale
FAU_GEN.1-NIAP-0410	SF.AU	This SFR is addressed completely by SF.AU as this function realizes the Security Audit mechanism of the TOE which logs all events required by FAU_GEN.1-NIAP-0410 and stores them into files in the environment.
FAU_GEN_EXP.2	SF.AU	This SFR is addressed by SF.AU as this function describes that the TOE stores the following information for every logged event: 1. Date and Time of the event 2. Type of Event 3. Identity of the user causing the event (if available) 4. ID of the object 5. Outcome (success or failure) of the event 6. For the rejection of a session additionally the reason for the rejection
FAU_SEL.1-NIAP-0407	SF.AU, SF.SM	This SFR is addressed by SF.AU as this Security Function allows in principle to include or exclude auditable events from being audited. However the administration is done using the Security Function SF.SM and SF.SM additionally ensures that only authorized administrators are allowed to use this management functionality.
FAU_STG_EXP.4	SF.AU, SF.SM	SF.AU allows the administrator to specify, what should happen in case the audit file are full. SF.AU is in these cases able to stop the TOE or to overwrite the old audit logs. SF.SM allows the admin to specify this action.
FDP_ACC.1	SF.AC	This SFR is completely addressed by SF.AC as this Security Function describes the Discretionary Access Control Mechanism as realized by the

		TOE which realizes Access Control based on the identity of the user and of the object.
FDP_ACF.1-NIAP-0407	SF.AC	This SFR is completely addressed by SF.AC as this Security Function describes the Discretionary Access Control Mechanism as realized by the TOE which invokes the same set of ordered rules as required by FDP_ACF.1-NIAP-0407
FDP_RIP.2	SF.SE	This SFR is completely addressed by the Security Function SF.SE as this Security Function ensures that any previous information content in memory is unavailable when the resource is allocated.
FIA_ATD.1	SF.I&A	This Security Function describes that the TOE maintains a security ID for each login on an instance level and each user on a database level and is able to associate these principals with their assigned roles in this way. It therefore completely realizes FIA_ATD.1.
FIA_UAU.2	SF.I&A	SF.I&A specifies that each user has to be successfully identified and authenticated before the TOE allows any other action on behalf of that user. It therefore completely realizes this SFR.
FIA_UAU.5	SF.I&A	SF.I&A describes that depending on the kind of the login the TOE is either reusing authentication results of the environment to authenticate a user or uses a Username/Password based mechanism to identify/authenticate a user. This completely realizes FIA_UAU.5
FIA_UID.2	SF.I&A	SF.I&A specifies that each user has to get successfully identified and authenticated before the TOE allows any other action on behalf of that user. It therefore completely realizes this SFR.
FMT_MOF.1	SF.SM	SF.SM provides the management function to start and stop the Security Audit and restricts the ability to use these functions to authorized administrators.
FMT_MSA.1	SF.SM	SF.SM provides the management function to manage all the security

		attributes and restricts the ability to use these functions to authorized administrators.
FMT_MSA_EXP.3	SF.AC	SF.AC specifies that if a new object is created only the owner(s) and the system administrator have access to this object. . Furthermore only users with a permission to parent objects (e.g. the schema or the database) have the same permission on the new object. In this way SF.AC realizes the policy of restrictive default values as required by FMT_MSA_EXP.3
FMT_MTD.1	SF.SM	SF.SM provides the management function to include or exclude events from being audited and restricts the ability to use these functions to authorized administrators.
FMT_REV.1(1)	SF.SM, SF.I&A	SF.SM provides the management functions to revoke security attributes associated with users and restricts the ability to use these functions to authorized administrators. SF.I&A specifies that changes to a SQL Server login are immediately applied while changes of a Windows Account name require a log of and log on of that user before they are applied.
FMT_REV.1(2)	SF.AC	SF.AC provides the functionality to revoke security attributes associated with objects and ensures that the revocation of attributes of these objects follows the DAC and all changes are applied immediately.
FMT_SMF.1	SF.SM	SF.SM provides all management functions required by FMT_SMF.1 and therefore completely realizes this SFR.
FMT_SMR.1	SF.SM	SF.SM maintains the role as required by FMT_SMR.1 and therefore completely realizes this SFR.
FPT_SEP_EXP.1	SF.I&A, SF.SE	SF.I&A ensures that a user is not allowed to perform any TSF mediated actions before he has been successfully identified and authenticated. Furthermore SF.SE ensures that for every operation requested by an authorized user at least one separate

		thread of the Operating System is started.
FPT_TRC_EXP.1	-	The TOE described in this ST does not comprise physically separated parts. So this SFR is trivially fulfilled in the context of the Application Note in paragraph 90 in chapter 5.1.5.2 of [PP].
FTA_MCS.1	SF.SE	SF.SE ensures that a user is only able to establish a session to the TOE as long as he has not reached the maximum number of concurrent sessions per user and by default enforces a maximum number of 5. In this way SF.SE completely realizes FTA_MCS.1
FTA_TAH_EXP.1	SF.SE	SF.SE stores the information about the attempts to establish a session required by FTA_TAH_EXP.1 and therewith completely realizes this SFR.
FTA_TSE.1	SF.SE	SF.SE ensures that a user is only able to establish a session to the TOE as long as no explicit deny statement has been specified for that user by an administrator. SF.SE completely realizes FTA_TSE.1.

8.9 Rationale for Assurance Measures

The Table 13 - Assurance Measures in chapter 6 shows how all assurance requirements were satisfied and that there is at least one assurance measure defined in the TOE Summary Specification to meet each of the security assurance requirements.

8.10 Rationale for PP Claims

This Security Target is compliant to the U.S. Government Protection Profile for Database Management Systems in Basic Robustness Environments, Version 1.1, 07.06.2006 ([PP]) because:

- All security objectives for the TOE from [PP] have been taken directly from [PP] into this ST without any refinement. Additionally the objective O.I&A has been introduced.
- All SFRs for the TOE have been taken directly from [PP] into this ST without any refinement. Only the permitted operations have been performed. Additionally SFRs for Identification and Authentication (FIA_UAU.2, FIA_UID.2 and FIA_UAU.5) and Prevention of audit data loss (FAU_STG_EXP.4.1) have been introduced.
- The following elements of SFRs contained uncompleted operations in [PP] which have been completed in this ST. The rest of the SFRs (except the additional ones mentioned before) has been taken from [PP] without any changes.

  - FAU_GEN.1.1-NIAP-0410
  - FAU_GEN.1.2-NIAP-0410 (Reference for table updated)
  - FAU_SEL.1.1-NIAP-0407
  - FDP_ACF.1.2-NIAP-0407
  - FDP_ACF.1.3-NIAP-0407
  - FDP_ACF.1.4-NIAP-0407
  - FIA_ATD.1.1
  - FMT_REV.1.2(1)
  - FMT_REV.1.2(2)
  - FMT_SMF.1.1
  - FMT_SMR.1.1
  - FTA_MCS.1.2
  - FTA_TSE.1.1
- Further FDP_RIP.2 has been used in this ST instead of FDP_RIP.1 as defined in the [PP].
- The SFR for the environment has been directly taken from [PP] without any changes. Additionally requirements for the non-IT environment (R.COMM and R.EVL) have been introduced.

9 Appendix

9.1 Definition for FIT_PPC_EXP

The additional family FIT_PPC_EXP (IT Environment PP Conformance) of the Class FIT (IT Environment) has been developed here to describe the IT security functional requirements of the environment.

The class FIT comprises only one family as seen in the following figure:

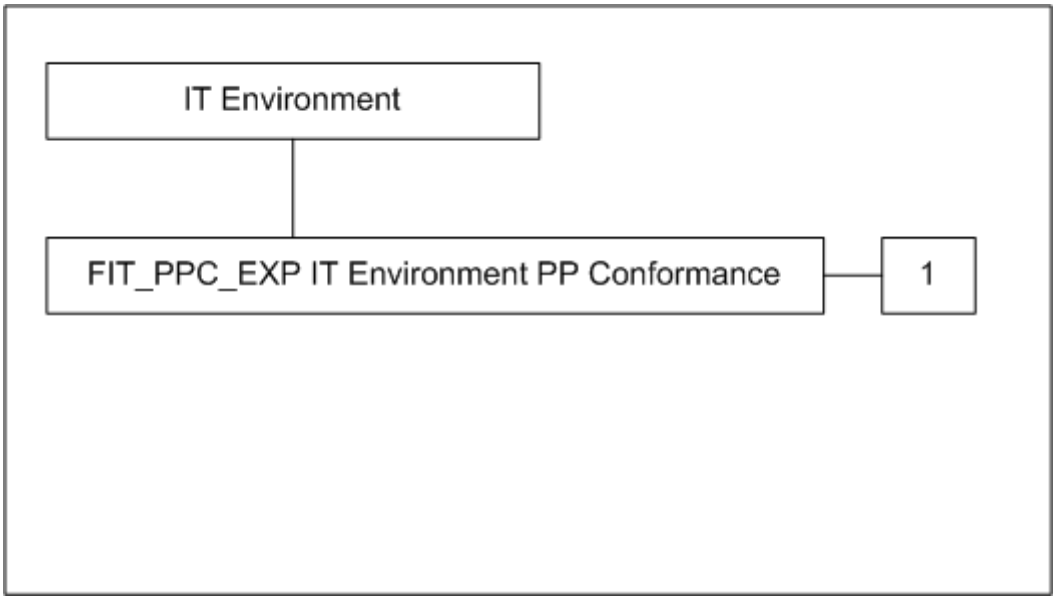

Figure 2: Class Structure for FIT

9.1.1 FIT_PPC_EXP (IT Environment Protection Profile Compliance)

Family Behaviour:

This family specifies the requirements that the environment of a TOE has to be compliant to a certain Protection Profile.

Component Levelling:

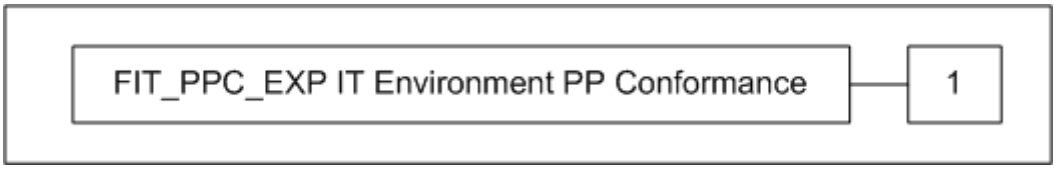

Figure 3: Component Levelling for FIT_PPC_EXP

FIT_PPC_EXP.1 shall be used to specify that the environment of a TOE has to be compliant to a certain Protection Profile.

Management: There are no management activities foreseen in this family.

Audit: There are no audit activities foreseen in this family.

IT Environment Protection Profile Compliance (FIT_PPC_EXP.1)

Hierarchical to:	No other components
FIT_PPC_EXP.1.1	The IT environment shall be compliant with the requirements of [Assignment: Protection Profile].
Dependencies:	No dependencies

9.2 Concept of Ownership Chains

Database Objects within the TOE are not always only passive objects. Some objects refer to other objects. This is especially true for Stored Procedures and Views. When multiple database objects access each other sequentially, the sequence is known as a chain. Although such chains do not independently exist, when the TOE traverses the links in a chain, the TOE evaluates access permissions on the constituent objects differently than it would if it were accessing the objects separately. These differences have important implications for managing security.

Ownership chaining enables managing access to multiple objects, such as multiple tables, by setting permissions on one object, such as a view. Ownership chaining also offers a slight performance advantage in scenarios that allow for skipping permission checks.

9.2.1 How Permissions Are Checked in a Chain

When an object is accessed through a chain, the TOE first compares the owner of the object to the owner of the calling object. This is the previous link in the chain. If both objects have the same owner, permissions on the referenced object are not evaluated. In the context of the Discretionary Access Control Mechanism this is not a circumvention of access control as the owner of an object always has complete control over his objects. So if one user is the owner of both objects, the calling object and the called object, the owner also would have direct access to both objects.

9.2.2 Example of Ownership Chaining

In the following illustration, the July2003 view is owned by Mary. She has granted to Alex permissions on the view. He has no other permissions on database objects in this instance. What happens when Alex selects the view?

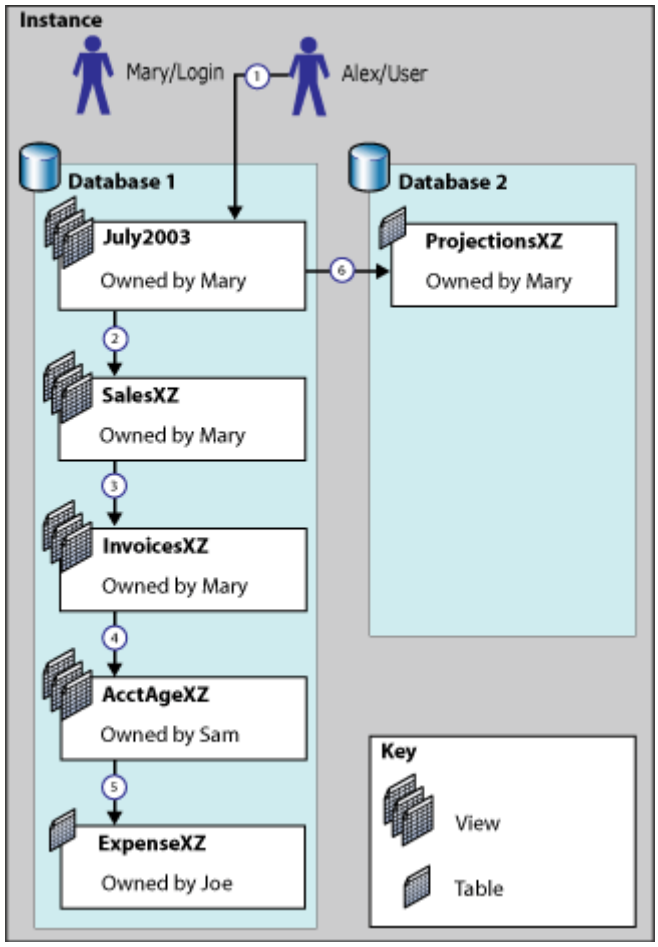

Figure 4: Concept of Ownership Chaining

Alex executes SELECT * on the July2003 view. The TOE checks permissions on the view and confirms that Alex has permission to select on it.

The July 2003 view requires information from the SalesXZ view. The TOE checks the ownership of the SalesXZ view. Because this view has the same owner (Mary) as the view that calls it, permissions on SalesXZ are not checked. The required information is returned.

The SalesXZ view requires information from the InvoicesXZ view. The TOE checks the ownership of the InvoicesXZ view. Because this view has the same owner as the previous object, permissions on InvoicesXZ are not checked. The required information is returned. To this point, all items in the sequence have had one owner (Mary). This is known as an unbroken ownership chain.

The InvoicesXZ view requires information from the AcctAgeXZ view. The TOE checks the ownership of the AcctAgeXZ view. Because the owner of this view is different from the owner of the previous object (Sam, not Mary), full information about permissions on this view is retrieved. If the AcctAgeXZ view has permissions that allow access by Alex, information will be returned.

The AcctAgeXZ view requires information from the ExpenseXZ table. The TOE checks the ownership of the ExpenseXZ table. Because the owner of this table is different from the owner of the previous object (Joe, not Sam), full information about permissions on this table

is retrieved. If the ExpenseXZ table has permissions that allow access by Alex, information is returned.

When the July2003 view tries to retrieve information from the ProjectionsXZ table, the TOE first checks to see whether cross-database chaining is enabled between Database 1 and Database 2. If cross-database chaining is enabled, the TOE will check the ownership of the ProjectionsXZ table. Because this table has the same owner as the calling view (Mary), permissions on this table are not checked. The requested information is returned.

9.3 References

The following documentation was used to prepare this ST:

[CC_PART2]	Common Criteria for Information Technology Security Evaluation – Part 2: Security functional requirements, dated August 2005, version 2.3, CCIMB-2005-08-002
[CC_PART3]	Common Criteria for Information Technology Security Evaluation – Part 3: Security assurance requirements, dated August 5005, version 2.3, CCIMB-2005-08-003
[CEM]	Common Evaluation Methodology for Information Technology Security – Evaluation Methodology, dated August 2005, version 2.3, CCIMB-2005-08-004
[PP]	U.S. Government Protection Profile for Database Management Systems in Basic Robustness Environments, Version 1.1, 07.06.2006
[CIM]	Consistency Instruction Manual for Development of US Government Protection Profiles for Use in Basic Robustness Environments, Version 3.0 (CIM)
[TSQL]	http://msdn.microsoft.com/library/default.asp?url=/library/en-us/acdata/ac_oview_4pcx.asp
[WIN_ST]	Microsoft Windows 2003/XP Security Target, Version 1.0. 28.09.2005, Microsoft Corporation
[WIN_VR]	National Information Assurance Partnership, Common Criteria Evaluation and Validation Scheme Validation Report Microsoft Windows 2003 Server and XP Workstation Report Number: CCEVS-VR-05-0131 Dated: November 6, 2005 Version: 1.1
[WIN_PP]	Controlled Access Protection Profile, Version 1.d, NSA, October, 8th, 1999

9.4 Glossary and Abbreviations

9.4.1 Glossary

The following abbreviations are used in this Security Target:

Abbreviation	Definition
Authorized Administrators	This term refers to a group of users which comprise the “sysadmin” (sa) and any user who is allowed to perform a management operation because the permission has been granted to him within the DAC either by assigning him to a role with administrator permissions or by granting him the possibility to perform an administrative operation explicitly.
DAC	Discretionary Access Control is a mechanism to limit the access of users to objects based on the ID of the user, the ID of the object and a set of access control rules.
DBMS	A DBMS is a computerized repository that stores information and allows authorized users to retrieve and update that information.
Named Pipe	Method for inter process communication
Object	An object within the TOE contains data and can be accessed by subjects. However in the TOE an object is not necessarily only a passive entity as some objects refer to other objects.
OC	Ownership Chaining. Explained in chapter 9.2 in more detail.
SQL	The Structured Query Language is a language which can be used to create, modify and retrieve data from a DBMS.
SQL Server	SQL Server is a product of Microsoft to which the TOE belongs.
TDS	Tabular Data Stream is a data format which is used for communication with the TOE.
T-SQL	Extension of the SQL language in order to support control flow, variables, user authentication and various other functions. See also http://msdn.microsoft.com/library/default.asp?url=/library/en-us/acdata/ac_oview_4pcx.asp

9.4.2 Abbreviations

The following abbreviations are used in this Security Target:

Abbreviation	Definition
ACL	Access Control List
CC	Common Criteria
CCEVS	Common Criteria Evaluation and Validation Scheme
CEM	Common Evaluation Methodology
CIM	Consistency Instruction Manual
DAC	Discretionary Access Control
DBMS	Database Management System
EAL	Evaluation Assurance Level
ETL	Extract, Transform, Load
IT	Information Technology
MOM	Microsoft Operations Manager
MS	Microsoft
NIAP	National Information Assurance Partnership
NSA	National Security Agency
OC	Ownership Chaining
ODS	Open Data Services
OLAP	Online analytical processing
OS	Operating System
OSP	Organizational Security Policy
PP	Protection Profile
sa	System administrator
SAR	Security Assurance Requirement
SFR	Security Functional Requirement
SID	Security ID
SMS	System Management Server
SOF	Strength of Function
SQL	Structured Query Language
ST	Security Target
TDS	Tabular Data Stream
TOE	Target of Evaluation
TSC	TSF Scope of Control
TSF	TOE Security Function

Abbreviation	Definition
T-SQL	Transact SQL