Microsoft® SQL Server® 2008 Database Engine
Common Criteria Evaluation

Security Target

SQL Server 2008 Team

Author: Roger French

Version: 1.2

Date: 2009-01-23

Abstract

This document is the Security Target (ST) for the Common Criteria certification of the database engine of Microsoft® SQL Server® 2008.

Keywords

CC, ST, Common Criteria, SQL, Security Target

This page intentionally left blank

Table of Contents

List of Tables

List of Figures

1 ST Introduction

This chapter presents Security Target (ST) and TOE identification information and a general 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, Security Problem Definition)

b) A set of security objectives and a set of security requirements to address the security problem (chapters 4 and 6, Security Objectives and IT Security Requirements, respectively).

c) The IT security functions provided by the TOE that meet the set of requirements (chapter 7, TOE Summary Specification).

1.1 ST and TOE Reference

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

ST Title:	**Microsoft SQL Server 2008 Database Engine Common Criteria Evaluation Security Target**
ST Version:	1.2
Date:	2009-01-23
Author:	Roger French, Microsoft Corporation
Certification-ID:	BSI-DSZ-CC-0520
TOE Identification:	Database Engine of Microsoft SQL Server 2008 Enterprise Edition (English) x86 and x64 and its related guidance documentation ([AGD] and [AGD_ADD])
TOE Version:	10.0.1600.22
TOE Platform:	Windows Server 2008 Enterprise Edition (English) Version 6.0.6001
CC Identification:	Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 1 as of September 2006 for part I, revision 2 as of September 2007 for parts II and III, English version.
Evaluation Assurance Level:	EAL 1 augmented by ASE_OBJ.2, ASE_REQ.2 and ASE_SPD.1.
PP Conformance:	none
Keywords:	CC, ST, Common Criteria, SQL, Security Target

1.2 TOE Overview

The TOE is the database engine of SQL Server 2008. 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

A summary of the TOE security functions can be found in chapter 1.3.4. A more detailed description of the security functions can be found in chapter 7, TOE Summary Specification.

Please note that only the SQL Server 2008 database engine is addressed in this ST. Other related products of the SQL Server 2008 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 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.

1.3.1 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 2008. It provides a relational database engine providing mechanisms for Access Control, Identification and Authentication and Security Audit.

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

- SQL Server Replication: Data replication for distributed or mobile data processing applications and integration with heterogeneous systems
- 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.
- Integration Services: Microsoft Integration Services is a platform for building enterprise-level data integration and data transformations solutions.
- Management tools: The SQL Server platform 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).
- 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
- Other tools offered by the installation process: Full Text Search, Business Intelligence Development Studio, Client tools connectivity, Client tools backwards compatibility, Client tools SDK, SQL client connectivity SDK, Microsoft sync framework.

The TOE itself only comprises the database engine of the SQL Server 2008 platform which provides the security functionality as required by this ST. Any additional tools of the SQL Server 2008 platform interact with the TOE as a standard SQL client. The scope and boundary of the TOE will be described in the next chapter. Please refer to [AGD_ADD] for more information about the installation process of the TOE.

1.3.2 Physical Scope and Boundary of the TOE

The TOE is the database engine of the SQL Server 2008 and its related guidance documentation. This engine has been evaluated in two different configurations (x86 and x64) while the IA64 version of the database engine has not been evaluated.

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

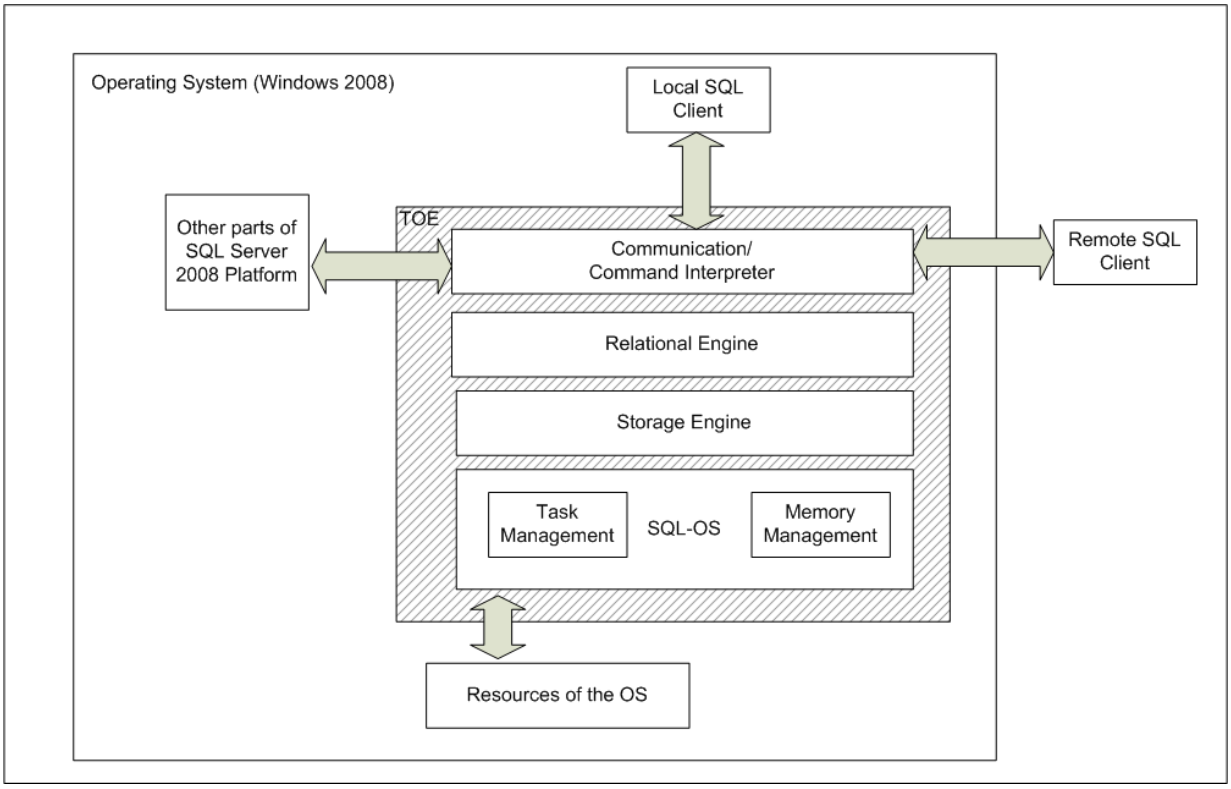

Figure 1: TOE

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.

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 2008 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 7 for more details)

Other parts of the SQL Server 2008 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 2008 Server Operating System and has the following hardware/software requirements:

Table 1: Hardware and Software Requirements

CPU	• Pentium III compatible at 1 GHz or faster (for the 32 bit edition) • AMD Opteron, AMD Athlon 64, Intel Xeon with Intel EM64T support, Intel Pentium IV with EM64T support at 1.4 GHz or faster [1]
RAM	512 MB
Hard Disk	Approx 1500 MB of free space
Other	DVD ROM drive, display at Super VGA resolution, Microsoft mouse compatible pointing device, keyboard
Software	Windows Server 2008 Enterprise Edition (in 64 or 32 bit), English version, version 6.0.6001 .NET Framework 3.5 SP 1 Windows Installer4.5

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

- SQL Server 2008 Books Online: This is the general guidance documentation for the complete SQL Server 2008 platform
- SQL Server Guidance Addendum / Installation / Startup: This document contains the aspects of the guidance that are specific to the evaluated configuration of SQL Server 2008

The website https://www.microsoft.com/sql/commoncriteria/2008/EAL1/default.mspx contains additional information about the TOE and its evaluated configuration. Also the guidance addendum that describes the specific aspects of the certified version can be obtained via this website. The guidance addendum extends the general guidance of SQL Server 2008 that ships along with the product in form of Books Online.

This website shall be visited before using the TOE.

1.3.3 Architecture of the TOE

The TOE which is described in this ST comprises one instance of the SQL Server 2008 database engine but has the possibility to serve several clients simultaneously.

1.3.4 Logical Scope and Boundary of the TOE

SQL Server 2008 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 2008 to the same machine.

The TOE comprises one instance of SQL Server 2008. 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".

[1] Please note that IA64 CPUs are not supported for the certified version of the database engine of SQL Server 2008

If more than one instance of SQL Server 2008 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 controls the access of users to user and TSF 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.

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**[2] for users for the cases where this is required by the TOE (The environment AND the TOE provide mechanisms for user authentication. See chapter 7.3 for more details).
- The environment has to provide **Time stamps** to be used by the TOE.
- The environment provides a **cryptographic** mechanisms for **hashing** of passwords

All these functions are provided by the underlying Operating System (Windows 2008 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

[2] Note that the TOE as well as the environment provides a mechanism for identification and authentication. Chapter 7 will describe this in more detail.

- TCP/IP

1.4 Conventions

For this Security Target the following conventions are used:

The CC allows several operations to be performed on functional requirements; *refinement*, *selection*, *assignment*, and *iteration* are defined in chapter C.4 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).

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.

2 Conformance Claims

2.1 CC Conformance Claim

This Security Target claims to be

- **CC Part 2 (Version 3.1, Revision 2, September 2007) extended** due to the use of the component FAU_STG.5.EXP
- **CC Part 3 (Version 3.1, Revision 2, September 2007) conformant** as only assurance components as defined in part III of [CC] have been used.

Further this Security Target claims to be conformant to the Security Assurance Requirements package EAL 1 augmented by ASE_OBJ.2, ASE_REQ.2 and ASE_SPD.1.

2.2 PP Conformance Claim

This Security Target does not claim compliance to any Protection Profile.

3 Security Problem Definition

This chapter describes

- the assets that have to be protected by the TOE,
- assumptions about the environment of the TOE,
- threats against those assets and
- organizational security policies that TOE shall comply with.

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 specifically includes:

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

3.2 Assumptions

The following table lists all the assumptions about the environment of the TOE.

Table 2 - 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	It is assumed that the TOE is installed on Windows Server 2008 Enterprise Edition and that this Operating System provides functionality for • Identification and authentication of users, • Access Control for Files, • Time stamps, • Audit Storage, • Hashing of passwords
A.PHYSICAL	It is assumed that appropriate physical security is provided for the server, on which the TOE is installed, considering 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.

3.3 Threats

The following table lists the threats against the assets, which are protected by the TOE and its environment.

Table 3 - Threats to the TOE

Threat	Description
T. ACCIDENTAL_ADMIN_ERROR	An administrator may incorrectly install or configure the TOE resulting in ineffective TSF data and therewith ineffective security mechanisms.
T.MASQUERADE	A user or process may claim to be another entity in order to gain unauthorized access to data or TOE resources.
T.TSF_COMPROMISE	A user or process may try to access (i.e. view, modify or delete) configuration data of the TOE. This could allow the user or process to gain knowledge about the configuration of the TOE or could bring the TOE into an insecure configuration in which the security mechanisms for the protection of the assets are not longer working correctly.
T.UNAUTHORIZED_ACCESS	A user may try to gain unauthorized access to user data for which they are not authorized according to the TOE security policy. Within the scope of this threat the user just tries to access assets, he doesn't have permission on, without trying to masquerade another user or circumventing the security mechanism in any other way.

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.

Table 4 – 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. This chapter describes the security objectives for the TOE and its operational environment.

4.1 Security Objectives for the TOE

This chapter identifies and describes the security objectives of the TOE.

Table 5 - Security Objectives for the TOE

Objective	Description
O.ADMIN_ROLE	The TOE will provide authorized administrators roles to isolate administrative actions. The TOE will provide administrators with the necessary information for secure management.
O.AUDIT_GENERATION	The TOE will provide the capability to detect and create records of security relevant events associated with users.
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.I&A	The TOE will provide a mechanism for identification and authentication of users.

4.2 Security Objectives for the operational Environment

The security objectives for the operational environment of the TOE are defined in the following table.

Table 6 - 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	The TOE shall be installed on Windows Server 2008 Enterprise Edition. This Operating System provides functionality for • Identification and authentication of users, • Access Control for Files, • Time stamps, • Audit Storage, • Hashing of passwords
OE.PHYSICAL	Physical security shall be provided for the server, on which the TOE will be installed, considering 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.
OE.AUDIT_REVIEW	The environment shall provide tools for the administrators to review the audit logs that are produced by the TOE.

4.3 Security Objectives Rationale

4.3.1 Overview

The following table summarizes the rationale for the security objectives.

Table 7 – Summary of Security Objectives Rationale

Threats, Assumptions, OSP / Security Objectives	O.ADMIN_ROLE	O.AUDIT_GENERATION	O.MANAGE	O.MEDIATE	O.I&A	OE.NO_EVIL	OE.NO_GENERAL_ PURPOSE	OE.OS	OE.PHYSICAL	OE.COMM	OE:AUDIT_REVIEW
T.ACCIDENTAL_ADMIN_ERROR	X										
T.MASQUERADE					X						
T.TSF_COMPROMISE			X								
T.UNAUTHORIZED_ACCESS				X	X						
P.ACCOUNTABILITY		X			X						X
P.ROLES	X										
A.NO_EVIL						X					
A.NO_GENERAL_PURPOSE							X				
A.OS								X			
A.PHYSICAL									X		
A.COMM										X	

Details are given in the following subchapters.

4.3.2 Rationale for TOE Security Objectives

Table 8 – 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_ROLE The TOE will provide administrators with the necessary information for secure management.	O.ADMIN_ROLE counters this threat by ensuring the TOE administrators have guidance that instructs them how to administer the TOE in a secure manner. Having this guidance and considering the assumption A.NO_EVIL mitigates the threat that an administrator might cause the TOE to be configured insecurely to an acceptable level.
T.MASQUERADE A user or process may claim to be another entity in order to gain unauthorized access to data or TOE resources.	O.I&A The TOE will provide a mechanism for identification and authentication of users.	O.I&A counters 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. In this way it is not possible for a user or process to masquerade as another entity and the threat is removed.
T.TSF_COMPROMISE A user or process may try to access (i.e. view, modify or delete) configuration data of the TOE. This could allow the user or process to gain knowledge about the configuration of the TOE or could bring the TOE into an insecure configuration in which the security mechanisms for the protection of the assets are not longer working correctly.	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 counters this threat as it defines that only authorized administrators shall be able to use the management functionality, provided by the TOE. In this way the threat is removed.
T.UNAUTHORIZED_ACCESS A user may try to gain unauthorized access to user data for which they are not authorized according to the	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-

TOE security policy. Within the scope of this threat the user just tries to access assets, he doesn't have permission on, without trying to masquerade another user or circumventing the security mechanism in any other way.		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. Together with O.I&A this mechanism ensures that no user can gain unauthorized access to data and in this way removes the threat.
	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 mechanism 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.
P.ACCOUNTABILITY The authorized users of the TOE shall be held accountable for their actions within the TOE.	O.AUDIT_GENERATION The TOE will provide the capability to detect and create records of security relevant events associated with users.	O.AUDIT_GENERATION addresses this policy by providing the authorized administrator with the capability of configuring the audit mechanism to record the actions of a specific user.
	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.

	OE.AUDIT_REVIEW	OE.AUDIT_REVIEW supports the policy for accountability as the environment of the TOE provides a means for audit review. Without this objective for the environment it would not be possible to review the audit logs that are produced by the TOE.
P.ROLES The TOE shall provide an authorized administrator role for secure administration of the TOE. This role shall be separate and distinct from other authorized users.	O.ADMIN_ROLE The TOE will provide authorized administrator roles to isolate administrative actions.	The TOE has the objective of providing authorized administrator roles for secure administration. In this way the policy P.ROLES is fulfilled. (by O.ADMIN_ROLE).

4.3.3 Rationale for environmental Security Objectives

The following table contains the rationale for the IT Environmental Objectives.

Table 9 – 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 computing or storage capabilities. This will protect the TSF data from malicious processes.
A.OS The TOE is installed on Windows Server 2008 Enterprise Edition. This Operating System provides functionality for • Identification and authentication of users, • Access Control for Files, • Time stamps, • Audit Storage, • Hashing of passwords	The TOE shall be installed on Windows Server 2008 Enterprise Edition. This Operating System provides functionality for • Identification and authentication of users, • Access Control for Files, • Time stamps, • Audit Storage, Hashing of passwords	The specific requirement on the Operating System ensures that the IT environment provides the necessary functionality for the operation of the TOE.
A.PHYSICAL It is assumed that appropriate physical security is provided for the server, on which the TOE is installed, considering the value of the stored, processed, and transmitted information.	OE.PHYSICAL Physical security shall be provided for the server, on which the TOE will be installed, considering 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.

5 Extended Component Definition

5.1 Definition for FAU_STG.5.EXP

This chapter defines the extended functional component FAU_STG.5.EXP (Administrable prevention of audit data loss) of the existing functional class FAU (Security audit).

This component was defined 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.

The family FAU_STG is extended by the new component FAU_STG.5.EXP as shown in the following figure:

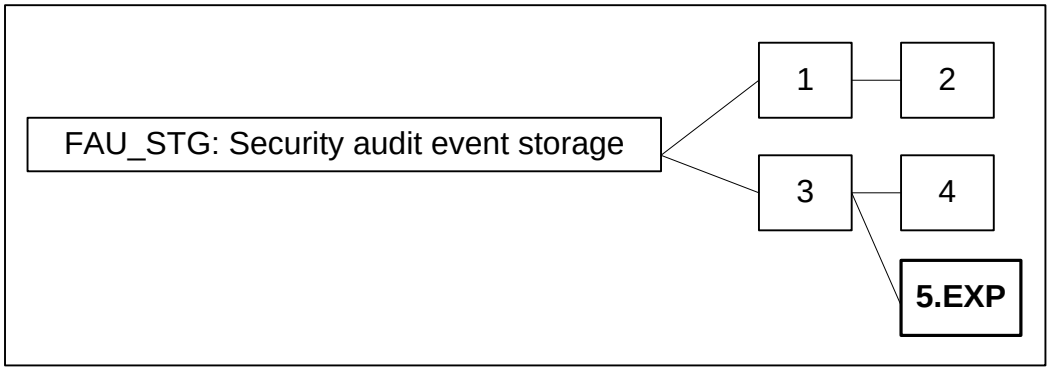

Figure 1: FAU_STG Component Levelling

FAU_STG.5.EXP Extended prevention of audit loss, specifies administrable actions in case the audit trail is full.

Management for FAU_STG.5.EXP:

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

a) maintenance (deletion, modification, addition) of actions to be taken in case of audit storage failure.

Audit for FAU_STG.5.EXP:

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

a) Basic: Actions taken due to potential audit storage failure.

FAU_STG.5.EXP Administrable prevention of audit data loss

	Hierarchical to:	FAU_STG.3 Action in case of possible audit data loss
	Dependencies:	FAU_STG.1 Protected audit trail storage
FAU_STG.5.EXP.1	The TSF shall **take one the following actions**: **[selection: *"ignore auditable events", "prevent auditable events, except those taken by the authorised user with special rights", "overwrite the oldest stored audit records", "stop the TOE",* [assignment: *other actions*]] as defined by [assignment: *authorised role*]** if the audit trail **is full.**	

6 IT Security Requirements

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

Common Criteria 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.

6.1 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 10 - TOE Security Functional Requirements

Class FAU: Security Audit	
FAU_GEN.1	Audit data generation
FAU_GEN.2	User identity association
FAU_SEL.1	Selective audit
FAU_STG.5.EXP	Administrable Prevention of audit data loss
Class FDP: User Data Protection	
FDP_ACC.1	Subset access control
FDP_ACF.1	Security attribute based access control
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.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

6.1.1 Class FAU: Security Audit

Audit data generation (FAU_GEN.1)

FAU_GEN.1.1 The TSF shall be able to generate an audit record of the following auditable events:

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

b) All auditable events for the *not specified* level of audit **listed in Table 11**; and

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

d) Use of special permissions (e.g., those often used by authorized administrators[3] to circumvent access control policies)]

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

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

b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [none].

[3] 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 8.3.1).

Table 11 - Auditable Events

Security Functional Requirement	Auditable Event(s)
FAU_GEN.1	None
FAU_GEN.2	None
FAU_SEL.1	All modifications to the audit configuration that occur while the audit collection functions are operating.
FDP_ACC.1	None
FDP_ACF.1	Successful requests to perform an operation on an object covered by the SFP.
FIA_ATD.1	None
FMT_MOF.1	None
FMT_MSA.1	None
FMT_MSA.3	None
FMT_MTD.1	None
FMT_REV.1(1)	Unsuccessful revocation of security attributes.
FMT_REV.1(2)	Unsuccessful revocation of security attributes.
FMT_SMF.1	Use of the management functions
FMT_SMR.1	Modifications to the group of users that are part of a role.
FAU_STG.5.EXP	Every modification 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 identity association (FAU_GEN.2)

FAU_GEN.2.1 For audit events resulting from actions of identified users[4], the TSF shall be able to associate each auditable event with the identity of the user that caused the event.

Selective audit (FAU_SEL.1)

FAU_SEL.1.1 The TSF shall **allow only the administrator** to select the set of audited events from the set of all auditable events based on the following attributes:

a) *user identity, object identity*,

b) [success of auditable security events, failure of auditable security events]

Administrable Prevention of audit data loss (FAU_STG.5.EXP)

FAU_STG.5.EXP.1 The TSF shall take one of the following actions:

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

as defined by [the administrator] if the audit trail is full.

6.1.2 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].

[4] Please note that the term user in this context refers to a user or a group of users.

Security attribute based access control (FDP_ACF.1)

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

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

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

6.1.3 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:[

- User identifier
- group memberships,
- login-type (SQL-Server login or Windows Account Name[5])
- For SQL-Server login: Hashed password].

[5] A windows account name may be a Windows user or a Windows group

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]

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,
- 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.

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

Static attribute initialization (FMT_MSA.3)

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

FMT_MSA.3.2 The TSF shall allow the [no role] to specify alternative initial values to override the default values when an object or information is created.

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 [*group membership*] associated with ~~**the**~~ *users* under the control of the TSF to [the authorized administrators and database users as allowed by the Discretionary Access Control policy].

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].

Revocation (FMT_REV.1(2))

FMT_REV.1.1(2) The TSF shall restrict the ability to revoke[6] [*Access Control Lists]* associated with ~~**the**~~ *objects* under the control of the TSF to [the authorized 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].

[6] In this context “revocation” refers to any change to an Access Control List that is associated with an object.

Specification of Management Functions (FMT_SMF.1)

FMT_SMF.1.1 The TSF shall be capable of performing the following 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
- Define the action to take in case the audit file is full]

Security roles (FMT_SMR.1)

FMT_SMR.1.1 The TSF shall maintain the roles:[

- 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 12 - Default Server Roles

Role	Descritpion
sysadmin	Members of the sysadmin fixed server role can perform any activity in the server. By default, all members of the Windows BUILTIN\Administrators group, the local administrator's group, are members of the sysadmin fixed server role.
Serveradmin	Members of the serveradmin fixed server role can change server-wide configuration options and shut down the server.
Securityadmin	Members of the securityadmin fixed server role manage logins and their properties. They can GRANT, DENY, and REVOKE server-level permissions. They can also GRANT, DENY, and REVOKE database-level permissions. Additionally, they can reset passwords for SQL Server logins.
Processadmin	Members of the processadmin fixed server role can end processes that are running in an instance of SQL Server.
Setupadmin	Members of the setupadmin fixed server role can add and remove linked servers.
Bulkadmin	Members of the bulkadmin fixed server role can run the BULK INSERT statement.
Diskadmin	The diskadmin fixed server role is used for managing disk files.
Dbcreator	Members of the dbcreator fixed server role can create, alter, drop, and restore any database.

Table 13 – Default Database Roles

Role	Granted Permission(s)
db_owner	Members of the db_owner fixed database role can perform all configuration and maintenance activities on the database, and can also drop the database.
db_securityadmin	Members of the db_securityadmin fixed database role can modify role membership and manage permissions. Adding principals to this role could enable unintended privilege escalation.
db_accessadmin	Members of the db_accessadmin fixed database role can add or remove access to the database for Windows logins, Windows groups, and SQL Server logins.
db_backupoperator	Members of the db_backupoperator fixed database role can back up the database.
db_ddladmin	Members of the db_ddladmin fixed database role can run any Data Definition Language (DDL) command in a database.

db_datawriter	Members of the db_datawriter fixed database role can add, delete, or change data in all user tables.
db_datareader	Members of the db_datareader fixed database role can read all data from all user tables.
db_denydatawriter	Members of the db_denydatawriter fixed database role cannot add, modify, or delete any data in the user tables within a database.
db_denydatareader	Members of the db_denydatareader fixed database role cannot read any data in the user tables within a database.

6.2 TOE Security Assurance Requirements

The assurance requirements for the TOE comprise all assurance requirements for EAL 1 as defined in [CC] augmented by ASE_OBJ.2, ASE_REQ.2 and ASE_SPD.1.

6.3 Security Requirements rationale

6.3.1 Security Functional Requirements rationale

The following table contains the rationale for the TOE Security Requirements.

Table 14 – Rationale for TOE Security Requirements

Objective	**Requirements Addressing the Objective**	**Rationale**
O.ADMIN_ROLE The TOE will provide authorized administrators roles to isolate administrative actions. The TOE will provide administrators with the necessary information for secure management.	FMT_SMR.1	The TOE will establish, at least, an authorized administrator role. 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.
	AGD_PRE.1	AGD_PRE.1 ensures the administrator has the information necessary to install the TOE in the evaluated configuration.
	AGD_OPE.1	AGD_OPE.1 mandates the developer provide the administrator and user with guidance on how to operate the

		TOE in a secure manner.
O.AUDIT_GENERATION The TOE will provide the capability to detect and create records of security relevant events associated with users.	FAU_GEN.1	FAU_GEN.1 defines the set of events that the TOE must be capable of recording. This requirement ensures that the administrator has the ability to audit any security relevant events that takes place in the TOE. This requirement also defines the information that must be contained in the audit record for each auditable event.
	FAU_GEN.2	FAU_GEN.2 ensures that the audit records associate a user and/or group identity with the auditable event.
	FAU_SEL.1	FAU_SEL.1 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.5.EXP	FAU_STG.5.EXP 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 TOE.
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.3	FMT_MSA.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.
	FDP_ACC.1 FDP_ACF.1	The access control policy of the TOE ensures that only authorized users (i.e. administrators) have access to the management functionality such that, for the TOE, the use of that functionality follows the same restrictions as the access to data.
	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 subjects and objects covered are defined by the TOE's policy.
	FDP_ACF.1	FDP_ACF.1 defines the security attribute used to provide access control to objects based on the TOE's access control policy.
O.I&A The TOE will provide a mechanism for identification and authentication of users.	FIA_ATD.1	FIA_ATD.1 defines the user attributes, necessary for authentication.
	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.

6.3.2 Rationale for satisfying all Dependencies

The following table contains the rationale for satisfying all dependencies of the Security Functional Requirements.

Table 15 – Functional Requirements Dependencies for the TOE

Requirement	Dependency	Satisfied
FAU_GEN.1	FPT_STM.1	This requirement is satisfied by the IT environment because the DBMS is a software only TOE.
FAU_GEN.2	FAU_GEN.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 (for SQL logins) or by the environment (For windows logins).
FAU_SEL.1	FAU_GEN.1 FMT_MTD.1	Satisfied
FAU_STG.5.EXP	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	Satisfied.
FDP_ACF.1	FDP_ACC.1 FMT_MSA.3	Satisfied
FIA_ATD.1	None	N/A
FIA_UAU.2	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) or by the environment (for windows logins).

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.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) The dependency to FIA_UID.1 is either fulfilled by the TOE (for SQL logins) or by the environment (For windows logins).

6.3.3 Rationale for Assurance Requirements

The "entry level" of EAL 1 has been chosen to gain an initial assurance that all required functionalities are correctly implemented by the TOE.

Facing the situation that future releases of the TOE may be certified at a higher assurance level it has been decided to augment the assurance level by ASE_OBJ.2, ASE_REQ.2 and ASE_SPD.1 to have a ST that could also be used for higher assurance levels.

7 TOE Summary Specification

This chapter presents an overview of the security functions implemented by the TOE.

7.1 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 (FMT_SMF.1):

- 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
- 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.

The TOE maintains a set of roles on the server level and on the database level as listed in Table 12 and 13. 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. (FIA_ATD.1, FMT_SMR.1)

Changes to logins that are preformed via the management functions are applied at the latest as soon as a new session for the login is established. (FMT_REV.1(1))

7.2 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 at most one instance “login”.

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. (FMT_MSA.3)

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. (FMT_REV.1(2)

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 8.1 in more detail the access is allowed

(FDP_ACC.1 and FDP_ACF.1)

As the access to management functions of the TOE is controlled by the same functionality as the access to user data this Security Function additionally ensures that the management functions are only available for authorized administrators. (FMT_MOF.1, FMT_MSA.1, FMT_MTD.1, FMT_REV.1(1))

7.3 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.[7]

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. (FIA_UAU.2, FIA_UID.2, FIA_UAU.5)

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.

[7] Windows authentication of users may be based on a username and password or alternative mechanisms. 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.

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. (FIA_UAU.2, FIA_UID.2, FIA_UAU.5)

7.4 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 on 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

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. Identity of the user causing the event (if available)
3. Type of the event
4. ID of the object
5. Outcome (success or failure) of the event

Furthermore each audit file contains an introduction with the list of events which are audited in the file. (FAU_GEN.1 and FAU_GEN.2)

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.

(FAU_STG.5_EXP)

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
- 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. (FAU_SEL.1)

8 Appendix

8.1 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.

8.1.1 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.

8.1.2 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?

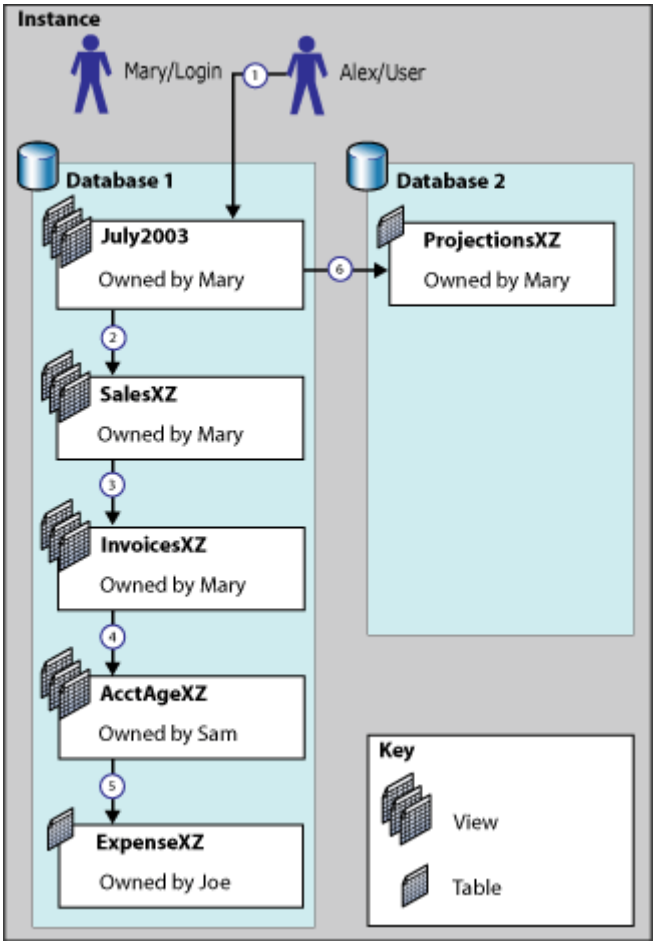

Figure 2: Concept of Ownership Chaining

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.

8.2 References

The following documentation was used to prepare this ST:

[CC]	Common Criteria for Information Technology Security Evaluation – Part 1: Introduction and general model, dated September 2006, version 3.1 R1 Part 2: Security functional requirements, dated September 2007, version 3.1, R2 Part 3: Security assurance requirements, dated September 2007, version 3.1, R2
[CEM]	Common Evaluation Methodology for Information Technology Security – Evaluation Methodology, dated September 2007, version 3.1 R2
[TSQL]	http://msdn.microsoft.com/library/default.asp?url=/library/en-us/acdata/ac_oview_4pcx.asp
[AGD]	Books online as of 2008-07-10
[AGD_ADD]	Microsoft SQL Server 2008 Database Engine Common Criteria Evaluation Guidance addendum, Version 1.2, 2008-11-26

8.3 Glossary and Abbreviations

8.3.1 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.
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.
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
Named Pipe	Method for inter process communication

8.3.2 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

Abbreviation	Definition
TSF	TOE Security Functionality
T-SQL	Transact SQL