Microsoft SQL Server 2022 on Linux
Database Engine
Common Criteria Evaluation (EAL2+)

Security Target

Author:	Wolfgang Peter (Microsoft Corporation)
Version:	1.2
Date:	2024-02-08

Abstract

This document is the Security Target (ST) for the Common Criteria certification of Microsoft SQL Server 2022 on Linux Database Engine Enterprise Edition x64 (English).

Keywords

CC, ST, Common Criteria, SQL, Security Target, DBMS, Database Management System

This page intentionally left blank

Table of Contents

List of Tables

Page

List of Figures

Page

1 ST Introduction

This chapter presents Security Target (ST) and TOE identification information and a general overview of the ST. A 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. A 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 functionality provided by the TOE that meets 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 2022 on Linux Database Engine Common Criteria Evaluation (EAL2+) Security Target
ST Version:	1.2
Date:	2024-02-08
Author:	Wolfgang Peter, Microsoft Corporation
TOE Identification:	Microsoft SQL Server 2022 on Linux Database Engine Enterprise Edition x64 (English) (and its related guidance documentation ([AGD] and [AGD_ADD]))
TOE Version:	16.0.4105.2
TOE Platform:	Ubuntu Linux Server 20.04, Red Hat Enterprise Linux 8, and SUSE Linux Enterprise Server v15
CC Identification:	Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 5 as of April 2017, English version ([CC]).
Evaluation Assurance Level:	EAL2 augmented by ALC_FLR.3
PP Conformance:	Collaborative Protection Profile (cPP) for Database Management Systems, Version 1.3, March 2023
Keywords:	CC, ST, Common Criteria, SQL, Security Target, DBMS, Database Management System

1.2 TOE Overview

The TOE is the database engine of SQL Server 2022 on Linux. SQL Server is a Database Management System (DBMS).

1.2.1 TOE Type

The type of the 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.

1.2.2 Usage and major security features

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 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: The TOE has the ability to restrict the access to security management functions only to authorized administrators.
- Access Control: The TOE provides the capability to restrict the access to the data and functionality to authorized users.
- Identification and Authentication: The TOE requires that each user must be successfully identified and authenticated before allowing any other actions. The TOE is also able to maintain a list of security attributes belonging to individual users.
- Security Audit: The TOE has the ability to generate and collect audit data regarding all security relevant events. Authorized administrators can also configure the audit system to exclude or include potentially auditable events to be audited based on a wide range of characteristics.
- Session Handling: The TOE provides the mechanisms to limit the possibilities of users to establish sessions with the TOE and maintain a separate execution context for every operation.

Note that only the SQL Server 2022 on Linux database engine is addressed in this ST. Other related products of the SQL Server 2022 platform, such as Analysis Services, 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.

1.2.3 Non-TOE Hardware/Software/Firmware

The TOE relies on functionality of the Operating System and has the following hardware/software requirements:

Aspect	Requirement
CPU	x64-compatible CPU, minimum dual-core at 1.4 GHz
RAM	Minimum 2 GB
File System	XFS or EXT4
Hard Disk	Minimum 6 GB
OS	Ubuntu Linux Server 20.04, or Red Hat Enterprise Linux 8, or SUSE Linux Enterprise Server v15
Software	OpenSSL Library

Aspect	Requirement
	For Active Directory authentication: realmd, Kerberos Client package, System Security Services Daemon (SSSD), Name Service Switch (NSS), Active Directory (AD)
Docker[1]	Docker Engine 1.8+

Table 1: Hardware and Software Requirements

1.3 TOE Description

This chapter provides context for the TOE evaluation by describing the evaluated configuration. The main purpose of this chapter is to bind the TOE in physical and logical terms. The scope and boundary of the TOE will be described in the next chapter.

1.3.1 Physical Scope and Boundary of the TOE

The TOE is the database engine of the SQL Server 2022 on Linux and its related guidance documentation. This engine is only available for x64 platforms. The TOE comprises one instance of the SQL Server 2022 on Linux database engine but has the possibility to serve several clients simultaneously.

The following figure shows the TOE (including its internal structure) and its immediate environment. In order to run on the Linux platform, the TOE comprises two components which are not part of the Windows release of SQL Server, namely the Platform Abstraction Layer (PAL) and the Linux Host Adapter which are described in more detail below.

[1] This dependency is optional. Docker Engine 1.8 will be required if the TOE is installed using the Docker container method. Please, see [AGD_ADD; section 3.2.3] for more detailed information

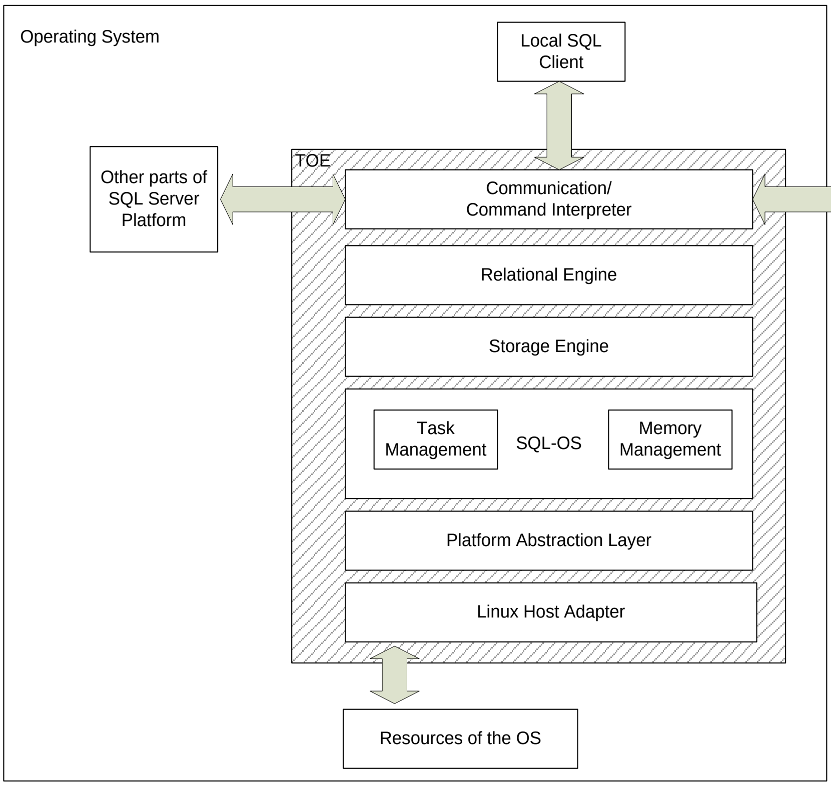

Figure 1: TOE Structure

As seen in Figure 1 the TOE internally comprises the following 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 Relational Engine allows compiling a subset of T-SQL statements into native code to create natively compiled Stored Procedures. The Visual C compiler used for this native compilation is not part of the TOE.

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 PAL and the underlying Linux 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 Linux Operating System.

The **Memory Management** 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 **Platform Abstraction Layer** provides a Windows-compatible interface to the upper layers of the TOE. It translates Windows DLL and Windows OS calls to Linux Host Adapter interface calls.

The **Linux Host Adapter** is the Linux executable that initializes the PAL so that the database engine can run on top of it. Furthermore, it provides access to Linux OS resources when requested by the PAL.

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

The Linux Operating System hosts the TOE. As the TOE is software only 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 functionality of the TOE (see chapter 7 for more details).

Other parts of the SQL Server 2022 Platform might be installed together with the TOE. The TOE is the central part of a complete DBMS platform, which realizes all security functionality 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 is downloadable as a Linux installable package (.rpm or .deb) from https://packages.microsoft.com.The complete path, filename and SHA-256 value for each installable package is as follows:

- Ubuntu 20.04

Path	https://packages.microsoft.com/ubuntu/20.04/mssql-server-2022/pool/main/m/mssql-server/mssql-server_16.0.4105.2-2_amd64.deb
Filename	mssql-server_16.0.4105.2-2_amd64.deb
SHA-256	040FF89B5B4FCEC12B827AD203EF4D2B19CFD9F08C939F61C34F0669225A50A4

- RHEL 8.0

Path	https://packages.microsoft.com/rhel/8/mssql-server-2022/Packages/m/mssql-server-16.0.4105.2-2.x86_64.rpm
Filename	mssql-server-16.0.4105.2-2.x86_64.rpm
SHA-256	9B868EEC2355C563DFF30C9ADA399A374B4B911C3A835BAAC5ED399F63D931D8

- SLES 15

Path	https://packages.microsoft.com/sles/15/mssql-server-2022/Packages/m/mssql-server-16.0.4105.2-2.x86_64.rpm
Filename	mssql-server-16.0.4105.2-2.x86_64.rpm
SHA-256	25AF132D01C7EB790CD1DA8288EE115500538B04FEE686EEE50EF90C10EA0536

The website https://www.microsoft.com/en-us/sql-server/data-security ([WEB]) (click on "View our Common Criteria certification" and a PDF document will be downloaded) 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 2022 on Linux. This website shall be visited before using the TOE.

The following guidance documents and supportive information belonging to the TOE can be obtained through [WEB] (the downloaded PDF contains the download links):

- Microsoft SQL Server 2022 Guidance Addendum: This document contains the aspects of the guidance that are specific to the evaluated configuration of SQL Server 2022 ([AGD_ADD]) and it is provided in .pdf format. It is the main document and should be downloaded first.
    - File name and version: SQL22_EAL2-L_AGD_ADD_1.1.pdf
    - SHA-256 value:
        - 1C97B50190B40BF64120BC2501046A6DDAACE043B949878C5C4F42F2B4FC760F
- Microsoft SQL Server 2022 Technical Documentation: This is the general guidance documentation for the complete SQL Server 2022 platform ([AGD]) and it is provided in .zip format.
    - File name: Offline-Book_SQL-Server-2022_1.0_2024-02-08.zip
    - SHA-256 value:
        - 48F2CFD270B05AA56945A2A0ADF6DBDB23FF667E6C52852FF42E95F178CC6909
- Microsoft SQL Server 2022 Permission Poster: This document contains all the possible permissions which apply to SQL Server 2022 ([PERM]) and it is provided in .pdf format.

NOTE: Although the permission poster refers to SQL Server 2017 is also applicable for the evaluated TOE.

- File name: Microsoft_SQL_Server_2017_and_Azure_SQL_Database_permissions_infographic.pdf
- SHA-256 value:
    - 4C2119AD0CB54B388D900590351FEB53758139EE6574B50EAB6BEF6192EC368B

- Installer Triggers Script: SQL script to install the necessary login triggers ([SCRIPTS]) and it is provided in .sql format.
    - File name: SQL22_L_Install_cc_triggers_1.0_2024-02-08.sql
    - SHA-256 value:
        - 3149F07F76F6D1F6381ED54611841660BFA231D7AC82A952292B2568BBC926D6

1.3.2 Logical Scope and Boundary of the TOE

The TOE comprises one instance of SQL Server 2022 on Linux database engine. It does not support running multiple instances on one machine. Within this ST the TOE is referenced either as "the TOE" or as "instance". The machine where the instance is running on is referenced as "server" or "DBMS-server".

The TOE provides the following set of security functionality:

- The **Access Control** function of the TOE controls the access of users to user and metadata 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 **Security Management** function allows authorized administrators to manage the behavior of the security functionality of the TOE.
- The **Identification and Authentication**[2] function of the TOE is able to identify and authenticate users.
- 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. Also the Memory Management functionality belongs to the area of Session Handling and 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 completely with new information.

Access to the complete functionality of the TOE is possible via a set of SQL-commands.

This set of commands is available via TCP/IP only.

1.3.3 Evaluated Configuration

The TOE is evaluated using the following six different server configurations:

- TOE running on Ubuntu Linux Server 20.04 with Azure Arc-extension enabled.
- TOE running on Ubuntu Linux Server 20.04 with Azure Arc-extension disabled.
- TOE running on Red Hat Enterprise Linux 8 with Azure Arc-extension enabled.
- TOE running on Red Hat Enterprise Linux 8 with Azure Arc-extension disabled.
- TOE running on SUSE Linux Enterprise Server v15 with Azure Arc-extension enabled.
- TOE running on SUSE Linux Enterprise Server v15 with Azure Arc-extension disabled.

1.4 Product Description

The TOE which is described in the above sections 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.

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
- Machine Learning Services: Machine learning functionality
- Full-Text and Semantic Extractions for Search: Search engine for database contents
- Data Quality Services: Data quality database objects

[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.

- PolyBaseQuery Service for External Data: Provides access to non-relational external data
- 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: 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 SQL Server Management Tools.
- 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.

The TOE itself only comprises the database engine of SQL Server 2022 on Linux which provides the security functionality as required by this ST. Any additional tools of the SQL Server 2022 platform interact with the TOE as a standard SQL client.

Moreover, the TOE needs that the operational environment provides some functionality for its correct operation. Those functionalities are the following:

- 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** mechanism for **hashing** of passwords.

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

1.5 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 [CC]. Each of these operations is used in this ST.

A **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** or in the case of deletions, by **~~crossed out 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 the cPP authors are denoted by *italicized text*, selections filled in this Security Target appear in square brackets with an indication that a selection is made, [selection:], and are not italicized.

The **assignment** operation is used to assign a specific value to an unspecified parameter, such as the length of a password. Assignments that have been made by the cPP authors are denoted by showing the value in square brackets, [assignment_value], assignments filled in this ST appear in square brackets with an indication that an assignment is to be made [assignment:].

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 security target authors to create their own requirements. Such requirements are termed ‘extended requirements’ and are permitted if the CC does not offer suitable requirements to meet the authors’ needs. Extended requirements must be identified and are required to use the CC class/family/component model in articulating the requirements. In this ST, extended requirements will be indicated with the tag “_EXT” between parenthesis 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 5, April 2017) extended** due to the use of the components FIA_USB_EXT.2 and FTA_TAH_EXT.1
- **CC Part 3 (Version 3.1, Revision 5, April 2017) 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 EAL2 augmented by ALC_FLR.3.

2.2 PP Conformance Claim

This Security Target claims to be conformant to:

- Collaborative Protection Profile for Database Management Systems, Version 1.3, March 13, 2023

As required by [PP], this Security Target claims exact conformance to [PP].

The product type of the TOE (see section 1.2.1) is consistent with the product type of the TOE specified in [PP] (both are database management systems (DBMS)). Since an exact conformance to [PP] is claimed no further conformance claim rationale is required.

3 Security Problem Definition

This chapter describes

- the external entities interacting with the TOE,
- 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 and Threat Agents

The threats given in section 3.3 refer to various threat agents and assets. The term "threat agent" is defined in CC Part 1.

The assets, mentioned in Table 1 below, are either defined in CC Part 1, or in glossary provided in section 8.3.1.

The terms "TSF data", "TSF" and "user data", are defined in CC Part 1. The terms "public objects" and "TOE resources" are given in the glossary provided in section 8.3.1.

3.2 Assumptions

The following table lists all the assumptions about the environment of the TOE. These assumptions have been directly taken from [PP] without any modification.

Assumption	Description
Physical aspects	
A.PHYSICAL	The operational environment is assumed to provide the TOE with appropriate physical protection such that the TOE is not subject to physical attack that may compromise the security and/or interfere with the platform's correct operation. This includes protection for the physical infrastructure on which the TOE depends for correct operation and hardware devices on which the TOE is executing.
Personnel aspects	
A.AUTHUSER	Authorized users possess the necessary authorization to access the information managed by the TOE in accordance with organization information access policies.
A.MANAGE	The TOE security functionality is managed by one or more competent, authorized administrators. The system administrative personnel are not careless, willfully negligent, or hostile, and will follow and abide by the instructions provided by the guidance documentation.
A.TRAINEDUSER	Authorized users are sufficiently trained to accomplish a task or a group of tasks within a secure IT environment by exercising control over their user data.
Procedural aspects	

Assumption	Description
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.PEER_FUNC_&_MGT	All external IT systems trusted by the TSF to provide TSF data or services to the TOE, or to support the TSF in the enforcement of security policy decisions are assumed to correctly implement the functionality used by the TSF consistent with the assumptions defined for this functionality and to be properly managed and operate under security policy constraints compatible with those of the TOE.
A.SUPPORT	Any information provided by a trusted entity in the IT environment and used to support the provision of time and date, information used in audit capture, user authentication, and authorization that is used by the TOE is correct and up to date.
Connectivity aspects	
A.CONNECT	All connections to and from remote trusted IT systems and between separate parts of the TSF are physically and/or logically protected within the TOE environment to ensure the integrity and confidentiality of the data transmitted and to ensure the authenticity of the communication end points.

Table 2: Assumptions

3.3 Threats

The following table identifies the threats to the TOE. These threats have been directly taken from [PP] without any modifications.

Threat	Description
T.ACCESS_TSFDATA	A user or a process may read or modify TSF data using functions of the TOE without being identified, authenticated and authorized.
T.ACCESS_TSFFUNC	A user or a process may use, manage or modify the TSF, bypassing the protection mechanisms of the TSF.
T.IA_USER	A user who has not successfully completed identification and authentication may gain unauthorized access to user data or TOE resources beyond public objects.
T.RESIDUAL_DATA	A user or a process acting on behalf of a user may gain unauthorized access to user or TSF data through reallocation of TOE resources from one user or process to another.

T.UNAUTHORIZED_ACCESS	An authenticated user or a process, in conflict with the TOE security policy, may gain unauthorized access to user data.

Table 3: Threats to the TOE

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.

Policy	Description
P.ACCOUNTABILITY	The authorized users of the TOE shall be held accountable for their actions within the TOE.
P.ROLES	Administrative authority to TSF functionality shall be given to trusted personnel and be as restricted as possible while supporting only the administrative duties the person has. This role shall be separate and distinct from other authorized users.
P.USER	Authority shall only be given to users who are trusted to perform the actions correctly and are permitted by the organization to access user data.

Table 4: Organizational Security Policies

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. The objectives have been directly taken from [PP].

Objective	Description
O.ADMIN_ROLE	The TOE shall provide roles that allow only authorized users to have access to administrative privileges that are specific to the role.
O.AUDIT_GENERATION	The TOE shall provide the capability to detect and create/generate records of security relevant events associated with users.
O.DISCRETIONARY_ACCESS	The TSF shall control access of subjects and/or users to named resources based on identity of the object, subject, or user. The TSF shall allow authorized users to specify for each access mode which users/subjects are allowed to access a specific named object in that access mode.

Objective	Description
O.I&A	The TOE shall ensure that users are authenticated before the TOE processes any actions that require authentication.
O.MANAGE	The TSF shall provide all the functions and facilities necessary to manage TOE security mechanisms, and shall restrict such management actions to authorized users.
O.RESIDUAL_INFORMATION	The TOE shall ensure that any information contained in a protected resource within its control is not inappropriately disclosed when the resource is reallocated.
O.TOE_ACCESS	The TOE shall provide functionality that controls a user's logical access to user data and to the TSF.

Table 5: Security Objectives for the TOE

4.2 Security Objectives for the Operational Environment

The security objectives for the Operational Environment are defined in the following table. The objectives for the environment have been directly taken from [PP] without any changes.

Objective	Description
OE.ADMIN	Those responsible for the TOE are competent and trustworthy individuals, capable of managing the TOE and the security of the information it contains.
OE.INFO_PROTECT	Those responsible for the TOE must establish and implement procedures to ensure that information is protected in an appropriate manner. In particular: • All network and peripheral cabling shall be approved for the transmittal of the most sensitive data transmitted over the link. Such physical links are assumed to be adequately protected against threats to the confidentiality and integrity of the data transmitted using appropriate physical and logical protection techniques. • DAC protections on security-relevant files (such as audit trails and authorization databases) shall always be set up correctly. • Users are authorized to access parts of the data managed by the TOE and are trained to exercise control over their own data.
OE.NO_GENERAL_ PURPOSE	There shall 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.

Objective	Description
OE.PHYSICAL	Those responsible for the TOE shall ensure that those parts of the TOE critical to enforcement of the security policy are protected from physical attack that might compromise IT security objectives. The protection shall be commensurate with the value of the IT assets protected by the TOE.

Table 6: Security Objectives for the Operational Environment

4.3 Security Objectives for the Operational IT Environment

The security objectives for the Operational IT Environment are defined in the following table. The objectives for the environment have been directly taken from [PP] without any changes.

Objective	Description
OE.IT_I&A	Any information provided by a trusted entity in the environment and used to support user authentication and authorization used by the TOE is correct and up to date.
OE.IT_TRUSTED_SYSTEM	External IT systems may be required by the TOE for the enforcement of the security policy. These external trusted IT systems shall be managed according to known, accepted, and trusted policies based on the same rules and policies applicable to the TOE, and are physically and shall be sufficiently protected from any attack that may cause those functions to provide false results.

Table 7: Security Objectives for the Operational IT Environment

4.4 Security Objectives Rationale

4.4.1 TOE Security Objectives Coverage

The following table maps the TOE security objectives to threats and OSPs. This information has been directly taken from [PP] without any changes:

Objective Name	SPD Coverage
O.ADMIN_ROLE	P.ACCOUNTABILITY P.ROLES T.ACCESS_TSFFUNC
O.AUDIT_GENERATION	P.ACCOUNTABILITY
O.DISCRETIONARY_ACCESS	T.IA_USER T.UNAUTHORIZED_ACCESS
O.I&A	P.ACCOUNTABILITY T.ACCESS_TSFFUNC T.ACCESS_TSFDATA T.IA_USER

Objective Name	SPD Coverage
O.MANAGE	P.USER T.ACCESS_TSFDATA T.ACCESS_TSFFUNC T.UNAUTHORIZED_ACCESS
O.RESIDUAL_INFORMATION	T.RESIDUAL_DATA
O.TOE_ACCESS	P.ACCOUNTABILITY P.ROLES P.USER T.ACCESS_TSFDATA T.ACCESS_TSFFUNC T.IA_USER T.UNAUTHORIZED_ACCESS

Table 8: Coverage of Security Objectives for the TOE

4.4.2 Rationale for TOE Security Objectives

The following table contains the rationale for the TOE security objectives. This rationale has directly been taken from [PP] without any changes.

Table 9: Rationale for TOE Security Objectives

Threat/Policy	TOE Security Objectives Addressing the Threat/Policy	Rationale
P.ACCOUNTABILITY The authorized users of the TOE shall be held accountable for their actions within the TOE.	O.ADMIN_ROLE The TOE shall provide roles that allow only authorized users to have access to administrative privileges that are specific to the role.	O.ADMIN_ROLE supports this policy by ensuring that the TOE provides a means of granting authorized administrators the privileges needed for secure administration.
	O.AUDIT_GENERATION The TOE shall provide the capability to generate records of security relevant events associated with users	O.AUDIT_GENERATION supports this policy by ensuring that audit records are generated to enable accountability.
	O.I&A The TOE shall ensure that users are authenticated before the TOE processes any actions that require authentication.	O.I&A supports this policy by requiring that each entity interacting with the TOE is properly identified and authenticated before allowing any action.

Threat/Policy	TOE Security Objectives Addressing the Threat/Policy	Rationale
	O.TOE_ACCESS The TOE shall provide mechanisms that control a user's logical access to user data and to the TSF.	O.TOE_ACCESS supports this policy by providing a mechanism for controlling user access.
P.USER Authority shall only be given to users who are trusted to perform the actions correctly and are permitted by the organization to access user data.	O.MANAGE The TSF shall provide all the functions and facilities necessary to manage TOE security mechanisms and shall restrict such management actions to authorized users.	O.MANAGE supports this policy by ensuring that the functions and facilities supporting secure management are in place.
	O.TOE_ACCESS The TOE shall provide mechanisms that control a user's logical access to user data and to the TSF.	O.TOE_ACCESS supports this policy by providing a mechanism for controlling user access.
	OE.ADMIN Those responsible for the TOE are competent and trustworthy individuals, capable of managing the TOE and ensuring the security of information it contains.	OE.ADMIN supports this policy by ensuring that only competent administrators are allowed to manage the TOE.
P.ROLES Administrative authority to TSF functionality shall be given to trusted personnel and be as restricted as possible supporting only the administrative duties the person has. This role shall be separate and distinct from other authorized users.	O.ADMIN_ROLE The TOE shall provide roles that allow only authorized users to have access to administrative privileges that are specific to the role.	O.ADMIN_ROLE supports this objective by providing roles that allow only authorized users access to administrative privileges.
	O.TOE_ACCESS The TOE shall provide mechanisms that control a user's logical access to user data and to the TSF	O.TOE_ACCESS supports this policy by controlling access to TSF functionality based on role.

Threat/Policy	TOE Security Objectives Addressing the Threat/Policy	Rationale
T.ACCESS_TSFDATA A user or a process may read or modify TSF data using functions of the TOE without being identified, authenticated and authorized.	O.I&A The TOE shall ensure that users are authenticated before the TOE processes any actions that require authentication.	O.I&A supports this policy by requiring that each entity interacting with the TOE is properly identified and authenticated before allowing any action the TOE is defined to provide to authenticated users only.
	O.MANAGE The TSF shall provide all the functions and facilities necessary to manage TOE security mechanisms, and shall restrict such management actions to authorized users.	O.MANAGE diminishes this threat since it ensures that functions and facilities used to modify TSF data are not available to unauthorized users.
	O.TOE_ACCESS The TOE shall provide mechanisms that control a user's logical access to user data and to the TSF.	O.TOE_ACCESS mitigates this threat by restricting TOE access.
T.ACCESS_TSFFUNC A user or a process may use, manage or modify the TSF, bypassing the protection mechanisms of the TSF..	O.ADMIN_ROLE The TOE will provide roles that allow only authorized users to have access to administrative privileges that are specific to the role.	O.ADMIN_ROLE mitigates this threat by restricting access to privileged actions.
	O.I&A The TOE shall ensure that users are authenticated before the TOE processes any actions that require authentication.	O.I&A mitigates this threat since the TOE requires successful authentication to the TOE prior to gaining access to any controlledaccess content

Threat/Policy	TOE Security Objectives Addressing the Threat/Policy	Rationale
	O.MANAGE The TSF shall provide all the functions and facilities necessary to manage TOE security mechanisms, and shall restrict such management actions to authorized users.	O.MANAGE mitigates this threat by ensuring that management functions are restricted to authorized users.
	O.TOE_ACCESS The TOE shall provide mechanisms that control a user's logical access to user data and to the TSF	O.TOE_ACCESS mitigates this threat by restricting TOE access.
T.IA_USER A user who has not successfully completed identification and authentication may gain unauthorized access to user data or TOE resources beyond public objects.	O.DISCRETIONARY_ACCESS The TSF shall control access of subjects and/or users to named resources based on identity of the object, subject, or user. The TSF shall allow authorized users to specify for each access mode which users/subjects are allowed to access a specific named object in that access mode.	O.DISCRETIONARY_ACCESS mitigates this threat by requiring that data, including user data stored with the TOE, is protected by discretionary access controls
	O.I&A The TOE shall ensure that users are authenticated before the TOE processes any actions that require authentication.	O.I&A mitigates this threat by requiring that each entity interacting with the TOE is properly identified and authenticated before allowing access beyond public objects.
	O.TOE_ACCESS The TOE shall provide mechanisms that control a user's logical access to user data and to the TSF.	O.TOE_ACCESS mitigates this threat by controlling logical access to user data and TSF data.

Threat/Policy	TOE Security Objectives Addressing the Threat/Policy	Rationale
T.RESIDUAL_DATA A user or a process acting on behalf of a user may gain unauthorized access to user or TSF data through reallocation of TOE resources from one user or process to another	O.RESIDUAL_INFORMATION The TOE shall ensure that any information contained in a protected resource is not inappropriately disclosed when the resource is reallocated.	O.RESIDUAL_INFORMATION mitigates this threat by ensuring that data is not improperly disclosed.
T.UNAUTHORIZED_ACCESS An authenticated user or a process, in conflict with the TOE security policy, may gain unauthorized access to user data.	O.DISCRETIONARY_ACCESS The TSF shall control access of subjects and/or users to named resources based on identity of the object, subject or user. The TSF shall allow authorized users to specify for each access mode which users/subjects are allowed to access a specific named object in that access mode.	O.DISCRETIONARY_ACCESS mitigates this threat by requiring that data, including TSF data, is protected by discretionary access controls.
	O.MANAGE The TSF shall provide all the functions and facilities necessary to manage TOE security mechanisms, and shall restrict such management actions to authorized users.	O.MANAGE mitigates this threat by ensuring that access to user data is restricted to authorized users.
	O.TOE_ACCESS The TOE shall provide mechanisms that control a user's logical access to user data and to the TSF.	O.TOE_ACCESS mitigates this threat by controlling logical access to user data and TSF data.

4.4.3 Security Objectives for Operational Environment Coverage

The following table maps the security objectives for the operational environment to assumptions, threats and OSPs. This information has been directly taken from [PP] without any changes:

Objective Name	SPD Coverage
OE.ADMIN	A.MANAGE P.USER

Objective Name	SPD Coverage
OE.INFO_PROTECT	A.AUTHUSER A.CONNECT A.MANAGE A.PHYSICAL A.TRAINEDUSER P.USER T.UNAUTHORIZED_ACCESS
OE.IT_I&A	A.SUPPORT
OE.IT_TRUSTED_SYSTEM	A.CONNECT A.PEER_FUNC_&_MGT
OE.NO_GENERAL_PURPOSE	A.NO_GENERAL_PURPOSE
OE.PHYSICAL	A.CONNECT A.PHYSICAL

Table 10: Coverage of Security Objectives for the Operational Environment

4.4.4 Rationale for Security Objectives for Operational Environment

The following table contains the rationale for the security objectives for the operational environment. This rationale has directly been taken from [PP] without any changes.

Table 11: Rationale for Security Objectives for the Operational Environment

Assumption	Environmental Objective Addressing the Assumption	Rationale
A.AUTHUSER Authorized users possess the necessary authorization to access the information managed by the TOE in accordance with organization information access policies	OE.INFO_PROTECT Those responsible for the TOE shall establish and implement procedures to ensure that information is protected in an appropriate manner. In particular: • All network and peripheral cabling shall be approved for the transmittal of the most sensitive data transmitted over the link. Such physical links are assumed to be adequately protected against threats to the confidentiality and integrity of the data transmitted using appropriate physical and logical protection techniques. • DAC protections on security-relevant files (such as audit trails and authorization databases) shall always be set up	OE.INFO_PROTECT supports the assumption by ensuring that users are authorized to access data managed by the TOE.

Assumption	Environmental Objective Addressing the Assumption	Rationale
	correctly. • Users are authorized to access parts of the data managed by the TOE and are trained to exercise control over their own data.	
A.CONNECT All connections to and from remote trusted IT systems and between separate parts of the TSF are physically and/or logically protected within the TOE environment to ensure the integrity and confidentiality of the data transmitted and to ensure the authenticity of the communication end points	OE.INFO_PROTECT Those responsible for the TOE shall establish and implement procedures to ensure that information is protected in an appropriate manner. In particular • All network and peripheral cabling shall be approved for the transmittal of the most sensitive data transmitted over the link. Such physical links are assumed to be adequately protected against threats to the confidentiality and integrity of the data transmitted using appropriate physical and logical protection techniques. • DAC protections on security-relevant files (such as audit trails and authorization databases) shall always be set up correctly. • Users are authorized to access parts of the data managed by the TOE and are trained to exercise control over their own data	OE.INFO_PROTECT supports the assumption by requiring that all network and peripheral cabling must be approved for the transmittal of the most sensitive data transmitted over the link. Such physical links are assumed to be adequately protected against threats to the confidentiality and integrity of the data transmitted using appropriate physical and logical protection techniques.
	OE.IT_TRUSTED_SYSTEM External IT systems may be required by the TOE for the enforcement of the security policy. These external trusted IT systems shall be managed according to known, accepted and trusted policies based on the same rules and policies applicable to the TOE, and shall be sufficiently	OE.IT_TRUSTED_SYSTEM supports the assumption by ensuring that external trusted IT systems implement the protocols and mechanisms required by the TSF to support the enforcement of the security policy..

Assumption	Environmental Objective Addressing the Assumption	Rationale
	protected from any attack that may cause those functions to provide false results.	
	OE.PHYSICAL Those responsible for the TOE shall ensure that those parts of the TOE critical to enforcement of the security policy are protected from physical attack that might compromise IT security objectives. The protection shall be commensurate with the value of the IT assets protected by the TOE.	OE.PHYSICAL supports the assumption by ensuring that appropriate physical security is provided within the domain..
A.SUPPORT Any information provided by a trusted entity in the IT environment and used to support the provision of time and date, information used in audit capture, user authentication, and authorization that is used by the TOE is correct and up to date.	OE.IT_I&A Any information provided by a trusted entity in the environment and used to support user authentication and authorization used by the TOE is correct and up to date.	OE.IT_I&A supports the assumption implicitly.
A.MANAGE The TOE security functionality is managed by one or more competent, authorized administrators. The system administrative personnel are not careless, willfully negligent, or hostile, and will follow and abide by the instructions provided by the guidance documentation.	OE.ADMIN Those responsible for the TOE are competent and trustworthy individuals, capable of managing the TOE and the security of information it contains.	OE.ADMIN supports the assumption by requiring that authorized administrators are competent, thereby ensuring that all the tasks are performed correctly and effectively.
	OE.INFO_PROTECT Those responsible for the TOE shall establish and implement procedures to ensure that information is protected in an appropriate manner. In particular • All network and peripheral cabling shall be approved for the transmittal of the most sensitive data transmitted over the link. Such physical links are assumed to be adequately protected	OE.INFO_PROTECT supports the assumption by ensuring that users are authorized to access the appropriate data, and are trained to exercise control.

Assumption	Environmental Objective Addressing the Assumption	Rationale
	against threats to the confidentiality and integrity of the data transmitted using appropriate physical and logical protection techniques. • DAC protections on security-relevant files (such as audit trails and authorization databases) shall always be set up correctly. • Users are authorized to access parts of the data managed by the TOE and are trained to exercise control over their own data	
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.	OE.NO_GENERAL_PURPOSE There shall 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.	OE.NO_GENERAL_PURPOSE The DBMS server must not include any general-purpose computing capabilities. This will protect the TSF data from malicious processes.
A.PEER_FUNC_&_MGT All external trusted IT systems trusted by the TSF to provide TSF data or services to the TOE, or to support the TSF in the enforcement of security policy decisions are assumed to correctly implement the functionality used by the TSF consistent with the assumptions defined for this functionality and to be properly managed and operate under security policy constraints compatible with those of the TOE	OE.IT_TRUSTED_SYSTEM External IT systems may be required by the TOE for the enforcement of the security policy. These external trusted IT systems shall be managed according to known, accepted, and trusted policies based on the same rules and policies applicable to the TOE, and shall be sufficiently protected from any attack that may cause those functions to provide false results.	OE.IT_TRUSTED_SYSTEM supports this assumption by ensuring that remote systems supporting the TOE are managed in a manner consistent with the security policies applicable to the TOE.

Assumption	Environmental Objective Addressing the Assumption	Rationale
A.PHYSICAL The operational environment is assumed to provide the TOE with appropriate physical protection such that the TOE is not subject to physical attack that may compromise the security and/or interfere with the platform's correct operation. This includes protection for the physical infrastructure on which the TOE depends for correct operation and hardware devices on which the TOE is executing..	OE.PHYSICAL Those responsible for the TOE shall ensure that those parts of the TOE critical to enforcement of the security policy are protected from physical attack that might compromise IT security objectives. The protection shall be commensurate with the value of the IT assets protected by the TOE.	OE.PHYSICAL supports this assumption by ensuring that the parts of the TOE critical to the enforcement of the security policy are protected from physical attack.
	OE.INFO_PROTECT Those responsible for the TOE shall establish and implement procedures to ensure that information is protected in an appropriate manner. In particular • All network and peripheral cabling shall be approved for the transmittal of the most sensitive data transmitted over the link. Such physical links are assumed to be adequately protected against threats to the confidentiality and integrity of the data transmitted using appropriate physical and logical protection techniques. • DAC protections on security-relevant files (such as audit trails and authorization databases) shall always be set up correctly. • Users are authorized to access parts of the data managed by the TOE and are trained to exercise control over their own data	OE.INFO_PROTECT supports the assumption by requiring that all network and peripheral cabling must be approved for the transmittal of the most sensitive data transmitted over the link. Such physical links are assumed to be adequately protected against threats to the confidentiality and integrity of the data transmitted using appropriate physical and logical protection techniques.
A.TRAINEDUSER	OE.INFO_PROTECT	OE.INFO_PROTECT

Assumption	Environmental Objective Addressing the Assumption	Rationale
Authorized users are sufficiently trained to accomplish a task or group of tasks within a secure IT environment by exercising control over their user data.	Those responsible for the TOE shall establish and implement procedures to ensure that information is protected in an appropriate manner. In particular • All network and peripheral cabling shall be approved for the transmittal of the most sensitive data transmitted over the link. Such physical links are assumed to be adequately protected against threats to the confidentiality and integrity of the data transmitted using appropriate physical and logical protection techniques. • DAC protections on security-relevant files (such as audit trails and authorization databases) shall always be set up correctly. • Users are authorized to access parts of the data managed by the TOE and are trained to exercise control over their own data	supports the assumption by ensuring that users are authorized to access parts of the data managed by the TOE and are trained to exercise control over their own data.
P.USER Authority shall only be given to users who are trusted to perform the actions correctly.	OE.ADMIN Those responsible for the TOE are competent and trustworthy individuals, capable of managing the TOE and the security of information it contains.	OE.ADMIN supports the policy by ensuring that the authorized administrators, responsible for granting authority to users, are trustworthy.
	OE.INFO_PROTECT Those responsible for the TOE must establish and implement procedures to ensure that information is protected in an appropriate manner. In particular: • All network and peripheral cabling must be approved for the transmittal of the most sensitive data transmitted over the link. Such physical	OE.INFO_PROTECT supports the policy by ensuring that users are authorized to access parts of the data managed by the TOE.

Assumption	Environmental Objective Addressing the Assumption	Rationale
	links are assumed to be adequately protected against threats to the confidentiality and integrity of the data transmitted using appropriate physical and logical protection techniques. • DAC protections on security-relevant files (such as audit trails and authorization databases) shall always be set up correctly. • Users are authorized to access parts of the data managed by the TOE and are trained to exercise control over their own data.	
T.UNAUTHORIZED_ACCESS A user may gain unauthorized access to user data for which they are not authorized according to the TOE security policy.	OE.INFO_PROTECT Those responsible for the TOE must establish and implement procedures to ensure that information is protected in an appropriate manner. In particular: • All network and peripheral cabling must be approved for the transmittal of the most sensitive data transmitted over the link. Such physical links are assumed to be adequately protected against threats to the confidentiality and integrity of the data transmitted using appropriate physical and logical protection techniques. • DAC protections on security-relevant files (such as audit trails and authorization databases) shall always be set up correctly. • Users are authorized to access parts of the data managed by the TOE and	OE.INFO_PROTECT diminishes the logical and physical threats by ensuring that the network and peripheral cabling are appropriately protected. DAC protections, when implemented correctly, support the identification of unauthorized access.

Assumption	Environmental Objective Addressing the Assumption	Rationale
	are trained to exercise control over their own data.	

5 Extended Components Definitions

5.1 Definition for FIA_USB_EXT.2

This chapter defines the extended functional component FIA_USB_EXT.2 Enhanced user-subject binding. The definition has been directly taken from [PP].

FIA_USB_EXT.2 is analogous to FIA_USB.1 except that it adds the possibility to specify rules whereby subject security attributes are also derived from TSF data other than user security attributes.

Component leveling

FIA_USB_EXT.2 is hierarchical to FIA_USB.1.

Management

See management description specified for FIA_USB.1 in [CC].

Audit

See audit requirement specified for FIA_USB.1 in [CC].

FIA_USB_EXT.2 Enhanced user-subject binding

Hierarchical to:	FIA_USB.1 User-subject binding
Dependencies:	FIA_ATD.1 User attribute definition

FIA_USB_EXT.2 .1

The TSF shall associate the following user security attributes with subjects acting on the behalf of that user: [assignment: list of user security attributes].

FIA_USB_EXT.2 .2

The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users: [assignment: rules for the initial association of attributes].

FIA_USB_EXT.2 .3

The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users: [assignment: rules for the changing of attributes].

FIA_USB_EXT.2 .4

The TSF shall enforce the following rules for the assignment of subject security attributes not derived from user security attributes when a subject is created: [assignment: rules for the initial association of the subject security attributes not derived from user security attributes].

5.2 Definition for FTA_TAH_EXT.1

This chapter defines the extended functional component FTA_TAH_EXT.1 TOE access information. The definition has been directly taken from [PP].

FTA_TAH_EXT.1 TOE access information provides the requirement for a TOE to make available information related to attempts to establish a session.

Component levelling

FTA_TAH_EXT.1 is not hierarchical to any other components.

Management: FTA_TAH_EXT.1

There are no management activities foreseen.

Audit: FTA_TAH_EXT.1

There are no auditable events foreseen.

FTA_TAH_EXT.1 TOE access information

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

FTA_TAH_EXT.1.1

Upon a session establishment attempt, the TSF shall store

a) the date and time of the session establishment attempt of the user.

b) the incremental count of successive unsuccessful session establishment attempt(s).

FTA_TAH_EXT.1.2

Upon successful session establishment, the TSF shall allow the date and time of

a) the previous last successful session establishment, and

~~b)~~ the last unsuccessful attempt to session establishment and the number of unsuccessful attempts since the previous last successful session establishment to be retrieved by the user.

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.

Class FAU: Security Audit	
FAU_GEN.1	Audit data generation
FAU_GEN.2	User identity association
FAU_SEL.1	Selective audit
Class FDP: User Data Protection	
FDP_ACC.1	Subset access control
FDP_ACF.1	Security attribute based access control
FDP_RIP.1	Subset residual information protection
Class FIA: Identification and Authentication	
FIA_ATD.1	User attribute definition
FIA_UAU.2	User authentication before any action
FIA_UID.2	User Identification before any action
FIA_USB_EXT.2	Enhanced user subject binding
Class FMT: Security Management	
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
Class FTA: TOE Access	
FTA_MCS.1	Basic limitation on multiple concurrent sessions

FTA_TAH_EXT.1	TOE access information
FTA_TSE.1	TOE session establishment

Table 12: TOE Security Functional Requirements

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 *minimum* level of audit **listed in Table 13: Auditable Events;** and

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

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

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 cPP/ST, [information specified in column three of Table 13: Auditable Events, below].

Security Functional Requirement	Auditable Event(s)	Additional Audit Record Contents
FAU_GEN.1	None	None
FAU_GEN.2	None	None
FAU_SEL.1	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	None
FDP_ACF.1	Successful requests to perform an operation on an object covered by the SFP.	None
FDP_RIP.1	None	None
FIA_ATD.1	None	None
FIA_UAU.2	Access denied by authentication mechanism	None
FIA_UID.2	Access denied by authentication mechanism	The user identity provided

Security Functional Requirement	Auditable Event(s)	Additional Audit Record Contents
FIA_USB_EXT.2	Unsuccessful binding of user security attributes to a subject (e.g. creation of a subject)	None
FMT_MSA.1	None	None
FMT_MSA.3	None	None
FMT_MTD.1	None	None
FMT_REV.1(1)	Unsuccessful revocation of security attributes.	Identity of individual attempting to revoke security attributes
FMT_REV.1(2)	Unsuccessful revocation of security attributes.	Identity of individual attempting to revoke security attributes
FMT_SMF.1	Use of the management functions.	Identity of the administrator performing these functions
FMT_SMR.1	Modifications to the group of users that are part of a role.	Identity of authorized administrator modifying the role definition
FTA_MCS.1	Rejection of a new session based on the limitation of multiple concurrent sessions	None
FTA_TAH_EXT.1	None	None
FTA_TSE.1	Denial of a session establishment due to the session establishment mechanism	Identity of the individual attempting to establish a session

Table 13: Auditable Events

User identity association (FAU_GEN.2)

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

Selective audit (FAU_SEL.1)

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

a) user identity;

b) [selection: object identity, event type, **success of auditable security events, failure of auditable security events**];

c) [assignment: none].

Application Note: ([PP, 6.1.2]) The intent of this requirement is to capture sufficient audit data to allow the administrators to perform their task, additional audit data may be captured.

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] **to objects** on [all subjects, all DBMS-controlled objects, and all operations among them].

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: [assignment:

- authorized users: user identity and/or group membership associated with the user[3],
- DBMS-controlled objects: object identity, access control rules for the object, ownership of object and parent object].

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [assignment:

- 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 The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: [assignment:

- 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 following rules: [assignment: no additional explicit denial rules].

Subset residual information protection (FDP_RIP.1)

FDP_RIP.1.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon the *allocation of the resource to* the following objects [assignment: objects that are related to or may be exposed through user sessions].

[3] The Discretionary Access Control policy is not enforced on system internal tasks that are not associated with an identified user.

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:

- [**Database user identifier and any associated group memberships;**
- **Security-relevant database roles; and**
- [assignment: login-type (SQL-Server login or Active Directory Authentication)]].

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.

Application Note: The TSF shall provide

- SQL Server Authentication and
- Access to Active Directory Authentication

to support user authentication.

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

- If the login is associated with an Active Directory user or group Active Directory 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 be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

Enhanced user-subject binding (FIA_USB_EXT.2)

FIA_USB_EXT.2.1 The TSF shall associate the following user security attributes with subjects acting on the behalf of that user: [assignment: the list of the security attributes as defined in FIA_ATD.1.1].

FIA_USB_EXT.2.2 The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users: [assignment: when a new user is created the public role and its associated privileges are assigned].

FIA_USB_EXT.2.3 The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users: [assignment: granting and revoking roles and privileges are effective immediately].

FIA_USB_EXT.2.4 The TSF shall enforce the following rules for the assignment of subject security attributes not derived from user security attributes when a subject is created: [assignment: none].

6.1.4 Class FMT: Security Management

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*] **all** the security attributes 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.

Application Note: ([PP, 6.4.1]) This requirement applies to new objects at the top-level (e.g., tables). When lower-level objects are created (e.g., rows, cells), these may inherit the permissions of the top-level objects by default. In other words, the permissions of the ‘child’ objects can take the permissions of the ‘parent’ objects by default.

FMT_MSA.3.2 The TSF shall allow **~~the~~** [no user] 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 [assignment: the following list of security attributes defined in FIA_ATD.1.1: associated group memberships and security-relevant database roles] associated with the *users* under the control of the TSF to [the authorized administrator].

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

Application Note: Security attributes “database user identifier” and “login-type” defined in FIA_ATD.1.1 cannot be revoked.

Revocation (DAC) (FMT_REV.1(2))

FMT_REV.1.1(2) The TSF shall restrict the ability to revoke [assignment: the following list of security attributes defined in FDP_ACF.1.1: access control rules for the object, ownership of the object and parent object] associated with the objects under the control of the TSF to [the authorized administrator] **and database users with sufficient privileges as allowed by the Discretionary Access Control policy**.

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

Application Note: Security attribute “object identity” defined in FDP_ACF.1.1 cannot be revoked since it is automatically generated.

Specification of Management Functions (FMT_SMF.1)

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

- Database configuration
- User and role management

[selection:

- Management of groups
- Adding or removing a database
- Revocation of security attributes
- Configuration of the maximum number of concurrent sessions
- Configuration of session establishment rules
- Configuration of TSF replication and consistency
- Configuration of TOE access information rules]

[assignment:

- 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 The TSF shall maintain the roles [authorized administrator 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.

Role	Description
sysadmin	Members of the sysadmin fixed server role can perform any activity in the server.
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.
diskadmin	The diskadmin fixed server role is used for managing disk files.

dbcreator	Members of the dbcreator fixed server role can create, restore and drop any database, and can alter their own databases.
public	Every SQL Server login belongs to the public server role. When a server principal has not been granted or denied specific permissions on a securable object, the user inherits the permissions granted to public on that object. Only assign public permissions on any object when you want the object to be available to all users. You cannot change membership in public.

Table 14: Default Server Roles

Role	Description
##MS_DatabaseConnector##	Members of the ##MS_DatabaseConnector## fixed server role can connect to any database without requiring a User-account in the database to connect to. To deny the CONNECT permission to a specific database, users can create a matching user account for this login in the database and then DENY the CONNECT permission to the database-user. This DENY permission will overrule the GRANT CONNECT permission coming from this role.
##MS_LoginManager##	Members of the ##MS_LoginManager## fixed server role can create, delete and modify logins. Contrary to the fixed server role securityadmin, this role does not allow members to GRANT privileges. It is a more limited role that helps to comply with the Principle of least Privilege.
##MS_DatabaseManager##	Members of the ##MS_DatabaseManager## fixed server role can create and delete databases. A member of the ##MS_DatabaseManager## role that creates a database, becomes the owner of that database, which allows that user to connect to that database as the dbo user. The dbo user has all database permissions in the database. Members of the ##MS_DatabaseManager## role don't necessarily have permission to access databases that they don't own.
##MS_ServerStateManager##	Members of the ##MS_ServerStateManager## fixed server role have the same permissions as the ##MS_ServerStateReader## role. Also, it holds the ALTER SERVER STATE permission, which allows access to several management operations, such as: DBCC FREEPROCCACHE, DBCC FREESYSTEMCACHE ('ALL'), DBCC SQLPERF()
##MS_ServerStateReader##	Members of the ##MS_ServerStateReader## fixed server role can read all dynamic management views (DMVs) and functions that are covered by VIEW SERVER STATE, and respectively has VIEW DATABASE STATE permission on any database on which the member of this role has a user account.
##MS_ServerPerformanceStateReader##	Members of the ##MS_ServerPerformanceStateReader## fixed server role

	can read all dynamic management views (DMVs) and functions that are covered by VIEW SERVER PERFORMANCE STATE, and respectively has VIEW DATABASE PERFORMANCE STATE permission on any database on which the member of this role has a user account. This is a subset of what the ##MS_ServerStateReader## server role has access to which helps to comply with the Principle of least Privilege.
##MS_ServerSecurityStateReader##	Members of the ##MS_ServerSecurityStateReader## fixed server role can read all dynamic management views (DMVs) and functions that are covered by VIEW SERVER SECURITY STATE, and respectively has VIEW DATABASE SECURITY STATE permission on any database on which the member of this role has a user account. This is a small subset of what the ##MS_ServerStateReader## server role has access to, which helps to comply with the Principle of least Privilege.
##MS_DefinitionReader##	Members of the ##MS_DefinitionReader## fixed server role can read all catalog views that are covered by VIEW ANY DEFINITION, and respectively has VIEW DEFINITION permission on any database on which the member of this role has a user account.
##MS_PerformanceDefinitionReader##	Members of the ##MS_PerformanceDefinitionReader## fixed server role can read all catalog views that are covered by VIEW ANY PERFORMANCE DEFINITION, and respectively has VIEW PERFORMANCE DEFINITION permission on any database on which the member of this role has a user account. This is a subset of what the ##MS_DefinitionReader## server role has access to.
##MS_SecurityDefinitionReader##	Members of the ##MS_SecurityDefinitionReader## fixed server role can read all catalog views that are covered by VIEW ANY SECURITY DEFINITION, and respectively has VIEW SECURITY DEFINITION permission on any database on which the member of this role has a user account. This is a small subset of what the ##MS_DefinitionReader## server role has access to which helps to comply with the Principle of least Privilege.

Table 15: Default predefined custom Server Roles

Note: The bulkadmin role may be listed by the TOE in operation, but it cannot be assumed by any user on Linux platforms.

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.

Role	Granted Permission(s)
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.

Table 16: Default Database Roles

6.1.5 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 [assignment: 5] sessions per user.

TOE access information (FTA_TAH_EXT.1)

FTA_TAH_EXT.1.1 Upon a session establishment, the TSF shall store

a. the date and time of the session establishment attempt of the user.
b. the incremental count of successive unsuccessful session establishment attempt(s).

FTA_TAH_EXT.1.2 Upon successful session establishment, the TSF shall allow the date and time of

a. the previous last successful session establishment, and
b. the last unsuccessful attempt to session establishment and the number of unsuccessful attempts since the previous last successful session establishment to be retrieved by the user.

TOE session establishment (FTA_TSE.1)

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

6.2 TOE Security Assurance Requirements

The assurance requirements for the TOE comprise all assurance requirements for EAL2 as defined in [CC] augmented by ALC_FLR.3.

Assurance Class	Assurance Component	Assurance Components Description
Development	ADV_ARC.1	Security architecture description
	ADV_FSP.2	Security-enforcing functional specification
	ADV_TDS.1	Basic design
Guidance Documents	AGD_OPE.1	Operational user guidance
	AGD_PRE.1	Preparative procedures
Life Cycle Support	ALC_CMC.2	Use of a CM system
	ALC_CMS.2	Parts of the TOE CM coverage
	ALC_DEL.1	Delivery procedures
	ALC_FLR.3	Flaw reporting procedures
Test	ATE_COV.1	Evidence of coverage
	ATE_FUN.1	Functional testing
	ATE_IND.2	Independent testing – sample
Vulnerability Assessment	AVA_VAN.2	Vulnerability analysis
Security Target evaluation	ASE_CCL.1	Conformance claims
	ASE_ECD.1	Extended components definition
	ASE_INT.1	ST introduction
	ASE_OBJ.2	Security objectives
	ASE_REQ.2	Derived security requirements
	ASE_SPD.1	Security problem definition
	ASE_TSS.1	TOE summary specification

Table 17: TOE Security Assurance Requirements

6.3 Security Requirements rationale

6.3.1 Security Functional Requirements rationale

The following table contains the rationale for the TOE Security Requirements. This rationale has been directly taken from [PP].

Table 18: Rationale for TOE Security Requirements

Objective	Requirements Addressing the Objective	Rationale
O.ADMIN_ROLE The TOE shall provide roles that allow only authorized users to have access to administrative privileges that are specific to the role.	FMT_SMR.1	The TOE will establish, at least, an authorized administrator role. Additional roles may also be specified.
O.AUDIT_GENERATION The TOE shall provide the capability to detect and create records of security relevant events associated with users..	FAU_GEN.1 FAU_GEN.2 FAU_SEL.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 ensures that the audit records associate a user and any associated group identity with the auditable event. FAU_SEL.1 allows the administrator to configure which auditable events will be recorded in the audit trail.
O.DISCRETIONARY_ACCESS The TSF shall control access of subjects and/or users to named resources based on identity of the object, subject or user. The TSF shall allow authorized users to specify for each access mode which users/subjects are allowed to access a specific named object in that access mode.	FDP_ACC.1 FDP_ACF.1	The TSF controls access to resources based on the subject and/or object security attributes.
O.I&A The TOE shall ensure that users are authenticated before the TOE processes any actions that require authentication.	FIA_ATD.1 FIA_UAU.2 FIA_UID.2 FIA_USB_EXT.2	FIA_UID.2 and FIA_UAU.2 ensure that only authorized users gain access to the TOE and its resources following identification and authentication. FIA_ATD.1 ensures that the security attributes used to determine access are defined and available to the support access control decisions.

Objective	Requirements Addressing the Objective	Rationale
		FIA_USB_EXT.2 ensures enforcement of the rules governing subjects acting on behalf of authorized users.
O.MANAGE The TSF shall provide all the functions and facilities necessary to manage TOE security mechanisms, and shall restrict such management actions to authorized users.	FMT_MSA.1 FMT_MSA.3 FMT_MTD.1 FMT_REV.1(1) FMT_REV.1(2) FMT_SMF.1 FMT_SMR.1	FMT_MSA.1 ensures that the ability to perform operations on security attributes is restricted to authorized administrators. FMT_MSA.3 ensures that default values used for security attributes are restrictive. FMT_MTD.1 ensures that the ability to include or exclude auditable events is restricted to authorized administrators. FMT_REV.1 restricts the ability to revoke attributes to the authorized administrator. FMT_SMF.1 identifies the management functions that are available to the authorized administrator. FMT_SMR.1 defines the specific security roles to be supported.
O.RESIDUAL_INFORMATION The TOE shall ensure that any information contained in a protected resource within its control is not inappropriately disclosed when the resource is reallocated.	FDP_RIP.1	FDP_RIP.1 ensures that the contents of resources are not available upon reallocation of the resource.
O.TOE_ACCESS The TOE shall provide mechanisms that control a user's logical access to user data and to the TSF.	FDP_ACC.1 FDP_ACF.1 FIA_ATD.1 FTA_MCS.1 FTA_TSE.1 FTA_TAH_EXT.1	FDP_ACC.1 and FDP_ACF.1 ensure that access between subjects and objects is controlled using security attributes. FIA_ATD.1 defines the security attributes for individual users. FTA_MCS.1 ensures that users are restricted to no more than a specified number of concurrent sessions. FTA_TSE.1 allows the TOE to restrict access to the TOE based on specified criteria.

Objective	Requirements Addressing the Objective	Rationale
		FTA_TAH_EXT.1 The TOE must be able to store and retrieve information about previous unauthorized login attempts and the number of 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. Access to this data is controlled and restricted such that a user may only access his or her own data.

6.3.2 Rationale for satisfying all Dependencies

The following table contains the rationale for satisfying all dependencies of the Security Requirements. This rationale has been directly taken from [PP]:

Table 19: Rationale for satisfying all dependencies

Requirement	Dependency	Satisfied
FAU_GEN.1	FPT_STM.1	This requirement is satisfied by the assumption on the IT environment, given in A.SUPPORT.
FAU_GEN.2	FAU_GEN.1 FIA_UID.1	This requirement is satisfied by FAU_GEN.1. This requirement is satisfied by FIA_UID.2 which is hierarchical to FIA_UID.1.
FAU_SEL.1	FAU_GEN.1 FMT_MTD.1	This requirement is satisfied by FAU_GEN.1. This requirement is satisfied by FMT_MTD.1.
FDP_ACC.1	FDP_ACF.1	This requirement is satisfied by FDP_ACF.1.
FDP_ACF.1	FDP_ACC.1 FMT_MSA.3	This requirement is satisfied by FDP_ACC.1. This requirement is satisfied by FMT_MSA.3.
FDP_RIP.1	None	N/A
FIA_ATD.1	None	N/A
FIA_UAU.2	FIA_UID.1	This requirement is satisfied by FIA_UID.2 which is hierarchical to FIA_UID.1.
FIA_UID.2	None	N/A

Requirement	Dependency	Satisfied
FIA_USB_EXT.2	FIA_ATD.1	This requirement is satisfied by FIA_ATD.1.
FMT_MSA.1	[FDP_ACC.1 or FDP_IFC.1] FMT_SMF.1 FMT_SMR.1	This requirement is satisfied by FDP_ACC.1. This requirement is satisfied by FMT_SMF.1. This requirement is satisfied by FMT_SMR.1.
FMT_MSA.3	FMT_MSA.1 FMT_SMR.1	This requirement is satisfied by FMT_MSA.1. This requirement is satisfied by FMT_SMR.1.
FMT_MTD.1	FMT_SMF.1 FMT_SMR.1	This requirement is satisfied by FMT_SMF.1. This requirement is satisfied by FMT_SMR.1.
FMT_REV.1(1)	FMT_SMR.1	This requirement is satisfied by FMT_SMR.1.
FMT_REV.1(2)	FMT_SMR.1	This requirement is satisfied by FMT_SMR.1.
FMT_SMF.1	None	N/A
FMT_SMR.1	FIA_UID.1	This requirement is satisfied by FIA_UID.2 which is hierarchical to FIA_UID.1.
FTA_MCS.1	FIA_UID.1	This requirement is satisfied by FIA_UID.2 which is hierarchical to FIA_UID.1.
FTA_TAH_EXT.1	None	N/A
FTA_TSE.1	None	N/A

6.3.3 Rationale for extended requirements

Table 20 presents the rationale for the inclusion of the extended functional requirements as already given in [PP].

Explicit Requirement	Identifier	Rationale
FIA_USB_EXT.2	Enhanced user-subject binding	Security attributes may be associated with a user to further restrict access or provide additional privileges
FTA_TAH_EXT.1	TOE Access Information	The TOE may make information related to attempts to establish a session available to users.

Table 20: Rationale for Extended Security Functional Requirements

6.3.4 Rationale for Assurance Requirements

The ‘entry level’ of EAL2 has been chosen to gain an initial assurance that all required functionalities are correctly implemented by the TOE. The additional use of ALC_FLR.3 is necessary in order to stay compliant to [PP].

7 TOE Summary Specification

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

7.1 Security Management (SF.SM)

This security functionality 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,
- 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.

The TOE maintains a set of roles on the server level and on the database level as listed in Table 14: Default Server Roles, **Table 15: Default predefined custom Server Roles** and Table 16: 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 (FIA_ATD.1, FIA_USB_EXT.2, FMT_SMR.1).

Changes to logins that are performed 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 TOE 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 the TOE controls the access of database users to objects of the database (e.g. to read or create a table).

Further the TOE is able to manage a user account completely within a database. In this case the user account in the database is associated with a login that is also contained in this database. The authentication then happens against this database.

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 initial ACL for a newly created object is always empty by default and cannot be overridden by any role (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 functionality additionally ensures that the management functions are only available for authorized administrators (FMT_MSA.1, FMT_MTD.1, FMT_REV.1(1)).

7.3 Identification and Authentication (SF.I&A)

This security functionality 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: Active Directory-based logins 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 environments provide Active Directory authentication.

Active Directory (AD) login names

These logins are associated with an AD user. For these logins the Kerberos protocol of the Active Directory in the environment is used to authenticate the user.[4] The TOE only verifies that the login has been successfully authenticated.

For these logins the Windows security identifier (SID) from the Active Directory account is used for identification of that login within the TOE. The mapping of SIDs to Linux user and group identifiers is performed by system services of the Linux environment (SSSD and NSS). Any permission is associated with that SID (FIA_UAU.2, FIA_UID.2, FIA_ATD.1, FIA_USB_(EXT).2).

SQL Server login names

SQL Server logins are not associated with an AD user but are maintained by the TOE itself. Logins exist on a server level (and users in databases can be associated with a login) and on a database level itself (for contained databases). 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 SHA2-512 hash function provided by OpenSSL as part of the Linux platform 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 (FIA_UAU.2, FIA_UID.2, FIA_ATD.1, FIA_USB_(EXT).2).

If the binding of a user security attribute to a subject fails at login (e.g., role membership), the login will also fail, and the failure of the login will be audited (FIA_USB_(EXT).2, FAU_GEN.1).

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,

[4] Kerberos authentication of users may be based on a username and password or alternative mechanisms. After successful authentication of a user Linux associates a used ID (UID) and a group ID (GID) with every user which represent the user and every group the user is a member of.

- 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 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. 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[5]:

1. Rollover

   The administrator specifies a maximum size per audit 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 audit file with a maximum size and the option to shut down the TOE on any audit error. When the maximum size of the audit 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. The event types to be audited are defined by Audit Action Groups which allow including or excluding classes of audit events on a server-level, database-level and audit-level (FAU_SEL.1).

[5] This information only covers the management function from requirement FMT_SMF.1, since this ST does not include any security functional requirement related to the audit storage.

7.5 Session Handling (SF.SE)

After a user attempting to establish a session has been successfully authenticated by SF.I&A this security functionality 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 (FTA_MCS.1).

As a second criterion 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 (FTA_TSE.1).

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. This information is available at the client interface at any time. It is not erased but only overwritten with updated values between sessions, i.e. the time of the last successful logon, the time of the last unsuccessful logon and the number of unsuccessful logon attempts between the last successful logon and the current successful logon are available until the user logs out (FTA_TAH_(EXT).1).

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 and which is related to the context of a user’s session (i.e. to which a user could gain access), this previous information will not be available for any user. To ensure this, the TOE either directly overwrites any memory that will be used for user sessions completely with new information or with a certain pattern. Before the previous information has been overwritten the resource is not available for any usage. Whenever data is written to or loaded from disc this is done page wise where a page has the size of 8 KB (FDP_RIP.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?

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.

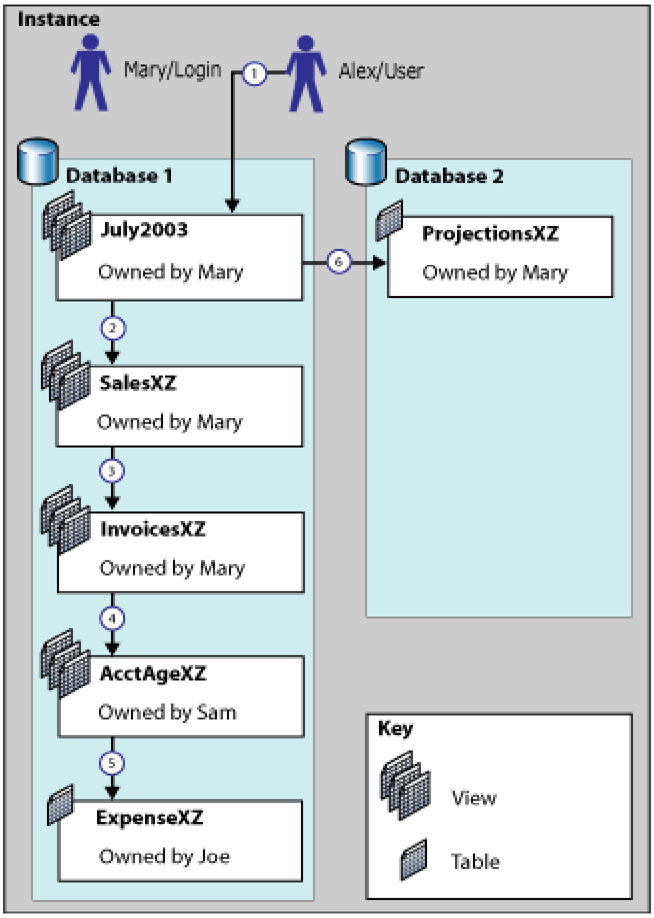

Figure 2: Concept of Ownership Chaining

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:

[AGD] Microsoft SQL Server 2022 Technical Documentation

[AGD_ADD] Microsoft SQL Server 2022 on Linux Database Engine Common Criteria Evaluation – Guidance Addendum

[CC] Common Criteria for Information Technology Security Evaluation

Part 1: Introduction and general model, dated April 2017, version 3.1 R5

Part 2: Security functional requirements, dated April 2017, version 3.1, R5

Part 3: Security assurance requirements, dated April 2017, version 3.1, R5

[PERM] Microsoft SQL Server 2022 Permission Poster

[PP] Collaborative Protection Profile (cPP) for Database Management Systems, Version 1.3, March 2023 ([PP])

[SCRIPTS] File containing a T-SQL script to install the CC logon triggers

[WEB] Website https://www.microsoft.com/en-us/sql-server/data-security (click on “View our Common Criteria certification” and a PDF document will be downloaded)

8.3 Glossary and Abbreviations

8.3.1 Glossary

The terms, definitions and abbreviations given [CC1] apply to this document as well as the additional terms, definitions and abbreviations listed in Glossary section of [PP]. Additionally, the following terms are used in this Security Target:

Term	Definition
Attacker	The term attacker refers to any individual (or technical entity) that is attempting to subvert the security functionality of the TOE. In this Security Target it is assumed that the attacker has an attack potential of “enhanced basic”.
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.
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.
User	The term user refers to technical entities (e.g. applications, other instances of the TOE) or human users (using a SQL-client) that are using the services of the TOE.

8.3.2 Abbreviations

The following abbreviations are used in this Security Target:

Abbreviation	Definition
ACL	Access Control List
AD	Active Directory
CC	Common Criteria

Abbreviation	Definition
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
OLAP	Online analytical processing
OS	Operating System
OSP	Organizational Security Policy
PAL	Platform Abstraction Layer
PP	Protection Profile
SAR	Security Assurance Requirement
SF	Security Functionality
SFR	Security Functional Requirement
SID	Security ID
SMS	System Management Server
SQL	Structured Query Language
ST	Security Target
TOE	Target of Evaluation
TSF	TOE Security Functionality
T-SQL	Transact SQL