



Microsoft® SQL Server® 2008R2 Database Engine Common Criteria Evaluation

Guidance Addendum

SQL Server 2008 R2 Team

Author:	Roger French
Version:	1.05
Date:	2011-12-08

Abstract

This document is the Guidance Addendum for the Common Criteria certification of the database engine of Microsoft® SQL Server® 2008 R2.

Keywords

CC, SQL, Common Criteria, Guidance Addendum

This page intentionally left blank

Table of Contents

	Page
1 INTRODUCTION.....	7
2 SCOPE OF THE EVALUATION.....	7
2.1 ASSUMPTIONS OF THE OPERATIONAL ENVIRONMENT.....	8
2.1.1 <i>Trained administrator</i>	8
2.1.2 <i>General purpose computing capabilities</i>	9
2.1.3 <i>Physical Protection</i>	9
3 INSTALLATION AND START-UP GUIDE.....	10
3.1 PREREQUISITES	10
3.1.1 <i>Hardware Prerequisites</i>	10
3.2 SOFTWARE PREREQUISITES	10
3.3 SQL SERVER 2008 R2 INSTALLATION.....	11
3.3.1 <i>Checking the integrity of the media</i>	11
3.3.2 <i>Installing the product</i>	12
3.3.3 <i>Installing SP1</i>	27
3.3.4 <i>Checking the version of the product</i>	28
3.3.5 <i>Format of version numbers</i>	28
3.3.6 <i>Enabling the certified version</i>	29
3.3.7 <i>Installing the logon triggers</i>	30
3.3.8 <i>Setting up the trace process</i>	31
3.3.9 <i>Basic verification of Security Functions</i>	31
4 SQL SERVER BOOKS ONLINE	33
5 GUIDANCE ADDENDUM	34
5.1 SQL SERVER STARTUP FLAGS	34
5.2 ADMINISTRATION INTERFACE	36
5.3 USER INTERFACE	37
5.4 SECURITY FUNCTIONS RELEVANT FOR ADMINISTRATION AND USE OF THE TOE	37
5.4.1 <i>Security Management</i>	38
5.4.2 <i>Session Handling</i>	46
5.4.3 <i>Access Control</i>	46
5.4.4 <i>Identification & authentication</i>	47
5.4.5 <i>Security Audit</i>	47
6 SQL SERVER TRACE	49
6.1 INFORMATION TO BE AUDITED	50
6.2 ROLE OF THE DEFAULT TRACE	51
6.3 THE "CC TRACE"	51
6.3.1 <i>Startup and shutdown of DBMS</i>	54
6.3.2 <i>Startup and shutdown of audit functions</i>	55
6.3.3 <i>Use of special permissions</i>	56

6.3.4	<i>Modifications to the audit configuration</i>	56
6.3.5	<i>Requests on operation</i>	57
6.3.6	<i>Unsuccessful revocation</i>	57
6.3.7	<i>Use of Management functions/Modifications of groups</i>	59
6.3.8	<i>Use of the authentication mechanism</i>	63
6.3.9	<i>Execution of Stored Procedures</i>	64
6.3.10	<i>User Error Message</i>	65
6.4	DEEPER AUDIT.....	66
6.5	FILTERING OF AUDIT AND PREVENTION OF AUDIT LOSS.....	67
6.6	SECURITY RELEVANT EVENTS.....	67
7	RECOMMENDATIONS AND REQUIREMENTS FOR SECURE ADMINISTRATION, CONFIGURATION AND USAGE	69
7.1	RECOMMENDATIONS/REQUIREMENTS ABOUT SECURITY AUDIT	69
7.2	RECOMMENDATIONS AND FURTHER INFORMATION ABOUT ACCESS CONTROL	69
7.3	RECOMMENDATIONS/REQUIREMENTS ABOUT IDENTIFICATION AND AUTHENTICATION (SECURE PASSWORDS)	71
7.4	OTHER RECOMMENDATIONS AND REQUIREMENTS	72
8	APPENDIX	75
8.1	STORED PROCEDURES	75
8.1.1	<i>sp_MSgetversion</i>	75
8.1.2	<i>xp_dirtree</i>	75
8.1.3	<i>xp_fileexist</i>	75
8.1.4	<i>xp_fixeddriives</i>	76
8.1.5	<i>xp_getnetname</i>	76
8.1.6	<i>xp_MSADEnabled</i>	76
8.1.7	<i>xp_qv</i>	77
8.1.8	<i>xp_instance_regread</i>	77
8.1.9	<i>xp_regread</i>	77
8.1.10	<i>sp_enable_sql_debug</i>	78
8.2	REFERENCES	79

List of Tables

	Page
Table 1: Assumptions on the operational environment	8
Table 2: Hash values for deliverables	11
Table 3: Commands to verify the integrity of SP1	27
Table 4: Entry Points into Books Online.....	33
Table 5: Startup Options for "sqlservr.exe"	35
Table 6: Commands to add and delete logins.....	39
Table 7: Commands to add and delete users	39
Table 8: Commands to add and delete users from database and server groups	40
Table 9: Commands to create and destroy database groups.....	40
Table 10: Commands to create, start and stop audit.....	41
Table 11: Commands to include and exclude auditable event	41
Table 12: Commands to grant, revoke and deny permissions.....	46
Table 13: Events to be audited	50
Table 14: Necessary audit events.....	54
Table 15: Important attributes of "Audit Server Starts and Stops" event.....	54
Table 16: Important attributes of "Audit Change Audit" event.....	55
Table 17: Important attributes of "Audit Server Alter Trace" event	56
Table 18: Important attributes of "Audit Object GDR" events	58
Table 19: Important attributes of "Audit Server Principal Management" event.....	59
Table 20: Important attributes of "Audit Database Principal Management" event.....	60
Table 21: Important attributes of "Audit Add Login to Server Role" event.....	61
Table 22: Important attributes of "Audit Add Member to DB Role" event.....	62
Table 23: Important attributes of "Audit Login" event	63
Table 24: Important attributes of "Audit Login Failed" event.....	64
Table 25: Important attributes of sp:starting and sp:completed	65
Table 26: Important attributes of "Audit User Error Message" event.....	66

List of Figures

	Page
Figure 1: Successful verification of integrity	12
Figure 2: Installing SQL Server 2008 R2 (I)	13
Figure 3: Installing SQL Server 2008 R2 (II)	14

Figure 4: Installing SQL Server 2008 R2 (III)	15
Figure 5: Installing SQL Server 2008 R2 (IV)	16
Figure 6: Installing SQL Server 2008 R2 (V)	17
Figure 7: Installing SQL Server 2008 R2 (VI)	18
Figure 8: Installing SQL Server 2008 R2 (VII)	19
Figure 9: Installing SQL Server 2008 R2 (VIII)	19
Figure 10: Installing SQL Server 2008 R2 (IX)	20
Figure 11: Installing SQL Server 2008 R2 (X)	21
Figure 12: Installing SQL Server 2008 R2 (XI)	22
Figure 13: Installing SQL Server 2008 R2 (XII)	23
Figure 14: Installing SQL Server 2008 R2 (XIII)	24
Figure 15: Installing SQL Server 2008 R2 (XIV)	25
Figure 16: Installing SQL Server 2008 R2 (XV)	26
Figure 17: Installing SQL Server 2008 R2 (XVI)	27
Figure 18: Basic verification results	32
Figure 19: Extract of permission hierarchy	70
Figure 20: Signature list of SP1	74

1 Introduction

This document has been created as part of the Common Criteria (CC) Evaluation of Database Engine of Microsoft SQL Server 2008 R2. It covers the specific aspects that shall be considered when operating SQL Server 2008 R2 in its certified version and extends the general guidance of the product given in Books Online. The document follows the following structure:

Chapter 2 of this document gives more details about the scope of the certification for SQL Server 2008 R2 and the assumptions, which have been made about the environment of the TOE.

Chapter 3 of this document describes the steps for the installation process of the database engine of SQL Server 2008 R2 in its certified version.

Chapter 4 introduces the concept of the SQL Server Books Online and provides the administrator and users with entry points for important aspects.

Chapter 5 contains the important aspects of the guidance, which are specific to the certified version of SQL Server 2008 R2.

Chapter 6 introduces the concept and the important aspects of the trace mechanism of SQL Server 2008 R2

Finally **chapter 7** gives requirements and recommendations for the secure operation of the TOE.

2 Scope of the evaluation

The Target of Evaluation (TOE), which has been addressed during this evaluation and certification process according to Common Criteria is one instance of the Database Engine of Microsoft SQL Server 2008 R2 Enterprise Edition (English) and Datacenter Edition (English), version 10.50.2500.0¹ x64 version and their related guidance documentation.

It should be noted that the certification results apply to both of the aforementioned editions of SQL Server. The descriptions within this document however are of generic nature so that they can be applied to both editions. The following descriptions will therefore not distinguish between the both editions.

This database engine is the core component of the SQL Server Platform.

The TOE has been defined to be one instance of the database engine as it realizes the complete set of security functions as described in [ST, chapter 6.1] including:

- Security Management,
- Access Control,
- Identification and Authentication
- Security Audit and

¹ Please note that the version number 10.50.2500.0 refers to SQL Server 2008 R2 plus an installed SP1.

- Session Handling

Additional information about the certification process and related documents can be obtained via [WEB].

The following chapter describes the assumptions, which have been made about the environment of the TOE during evaluation, and which therefore have to be addressed during the start-up and operation of the TOE. It further explains how these assumptions can be addressed.

2.1 Assumptions of the Operational Environment

According to [ST] the following assumptions apply to the environment of use of the TOE.

Assumption	Description
A.NO_EVIL	Administrators are non-hostile, appropriately trained, and follow all administrator guidance.
A.NO_GENERAL_PURPOSE	There are no general-purpose computing capabilities (e.g., compilers or user applications) available on DBMS servers, other than those services necessary for the operation, administration and support of the DBMS.
A.PHYSICAL	It is assumed that appropriate physical security is provided within the domain for the value of the IT assets protected by the TOE and the value of the stored, processed, and transmitted information.

Table 1: Assumptions on the operational environment

The following chapters provide more details about the requirements which result out of the several assumptions for the secure administration of the TOE.

2.1.1 Trained administrator

To address the assumption A.NO_EVIL, authorized administrators shall read and follow all guidance documentation.

Further this assumption requires appropriate training for the administrators.

It is assumed that the 'sa' (sa stands for system administrator and represents the administrative user that has the highest level of permissions) has a commensurate level of knowledge to a MCDBA. Therefore, it is recommended that the 'sa' receive formal DBA training on the level of a MCDBA (or equivalent).

It is the responsibility of 'sa' to ensure that all other authorized administrators have sufficient knowledge and skills for the scope of their administrative permissions.

2.1.2 General purpose computing capabilities

The administrator of the TOE shall not install and/or use any general computing software on the machine where the TOE has been or will be installed other than those services necessary for the operation, administration and support of the DBMS. Only applications or services indispensable for the operation or installation of the TOE shall be run. The installation of the TOE has to be performed on a virgin OS. Beside the installation of the TOE itself a SQL-client may be installed on the machine to be used for administration. Also the SQL Server Management Studio and Books Online which ship together with the TOE can be used for administration (see also [AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_4deptrbl/html/d2bade70-07cf-4d94-b5d2-88aecb538ed1.htm]).

However it should be noted that Management Studio has not been within the scope of evaluation. Specifically all functionality of the Graphical User Interface has not been evaluated. Thus – within the context of the evaluation Management Studio should just be seen as any other T-SQL client and the administrator shall ensure that the version of the client he is using is up to date and does not introduce any potential vulnerabilities.

Further other parts of the SQL Server 2008 R2 Platform may be installed as long as they are needed to support the administration and operation of the TOE.

For example the SQL Server Profiler may be used to review the audit logs (see also [AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_4deptrbl/html/3ad5f33d-559e-41a4-bde6-bb98792f7f1a.htm])

The administrator shall further consider that also other users with accounts on the local machine the SQL Server is running on may have the possibility to install and use general computing software. This scenario however, is also covered by the assumption described in this chapter. Therefore the administrator shall also ensure that other users do not have the possibility to install and/or use general computing capabilities or untrusted software.

One way to enforce this would e.g. be to have no accounts for non administrative users on the machine the SQL Server database engine is running on.

2.1.3 Physical Protection

It shall be ensured by the administrator that appropriate physical security is provided within the domain for the value of the IT assets protected by the TOE and the value of the stored, processed, and transmitted information. Physical protection of the wire can provide an adequate level of security for the information transmitted between the clients and the TOE as all connections from clients to the TOE are unencrypted per default. It has to be mentioned that the maximum level of protection the TOE can provide for the user data which is stored in it depends on the physical security of the machine where the TOE is installed. With physical access to this machine an attacker could easily gain complete access to the user data which is stored in the database.

3 Installation and Start-up Guide

This chapter provides instructions for a secure setup, installation, and configuration of the TOE. In addition, this chapter describes the prerequisites for the installation process.

3.1 Prerequisites

3.1.1 Hardware Prerequisites

According to [ST] a machine that meets at least following criteria has to be available:

- AMD Opteron, AMD Athlon 64, Intel Xeon with Intel EM64T support, Intel Pentium IV with EM64T support at 1.4 GHz or faster (Please note that IA64 CPUs are not supported for the certified version of the database engine of SQL Server 2008 R2)
- 1 gigabyte (GB) of RAM or more
- Approximately 1500 MB of available hard-disk space for the recommended installation
- DVD-ROM drive
- SVGA (1,024x768) or higher-resolution video adapter and monitor
- Microsoft Mouse or compatible pointing device
- keyboard

Please note that additional disc space will be required for the recommended trace processes (Up to 10 GB in its default configuration).

3.2 Software Prerequisites

Before the installation of the TOE can start the following Operating System and additional prerequisites have to be installed on the machine:

- Windows Server 2008 Enterprise Edition R2 (in x64), English version .NET Framework 3.5 SP 1²
- Windows Installer 4.5²

Further it is recommended to consider installing critical updates for those products before proceeding with the installation. However, it should be noted that any configuration of SQL Server that bases on a different configuration of the software prerequisites has not been considered during evaluation. In this context it should be noted that the installers for the .NET Framework and the Windows Installer do automatically receive updates if the machine is connected to the internet. In order to ensure that the exact version is installed the administrator shall therefore consider to disconnect the machine from the internet before installation.

² This will also be automatically installed by the installer of SQL Server if not already on the machine.

3.3 SQL Server 2008 R2 Installation

3.3.1 Checking the integrity of the media

It is assumed that the administrator has already successfully verified the integrity of the SQL Server 2008 R2 Guidance Addendum (this document) as described on [WEB]. This activity includes the secure download of the FCIV tool.

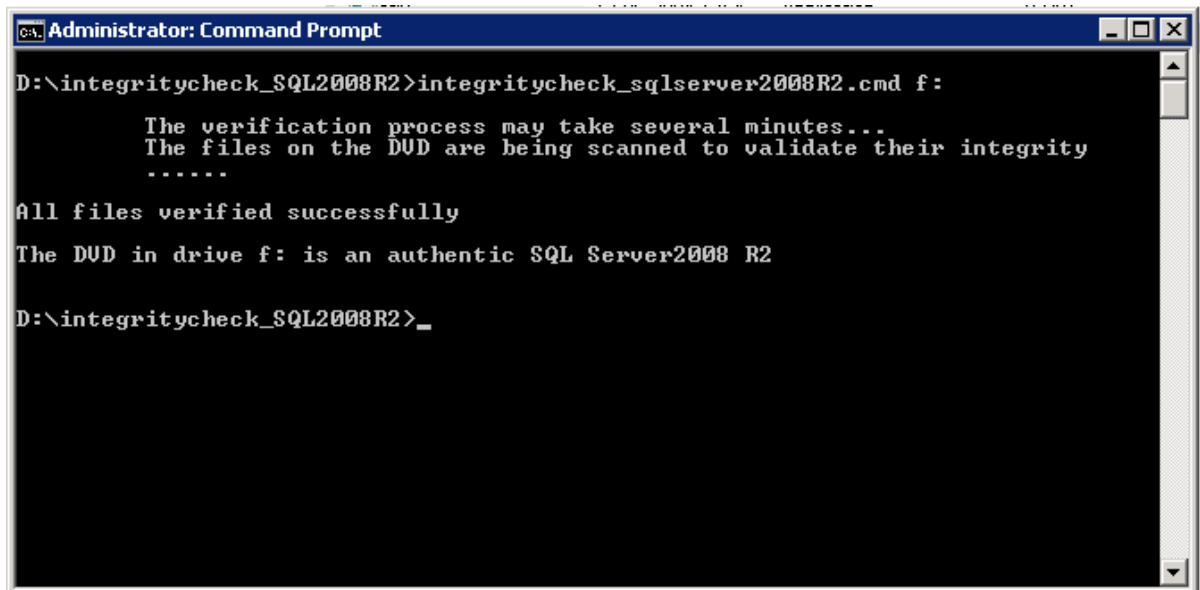
Before installing the product the administrator shall furthermore verify the integrity of the installation media and all other downloads. This verification shall be done as follows:

1. Download the following files from [WEB] to the folder that contains the FCIV tool:
 - Install_cc_triggers.sql
 - EAL4_trace.sql
 - integritycheck_SQL2008R2.zip
 - permission_hierarchy.zip
 - verification_script.zip
2. Open a command prompt and change to the directory where FCIV has been extracted. Run “fciv –sha1 <file>” for each downloaded file and compare the output hash with the following hashes:

File	SHA1 Hash
Install_cc_triggers.sql	01859d7d0c859b418896e403353a63d568b1b4d9
EAL4_trace.sql	08e205ebafce4157cb1480aa05259de8193af84a
integritycheck_SQL2008R2.zip	4cd81c3458a7e0f4e8858d4527ee98e59d697a01
permission_hierarchy.zip	578bf0aa2fb56e113118b6e00ed2aec75fe95a8f
verification_script.zip	cebcbbeb2e0a4b78ae1d14739351496af376704cc

Table 2: Hash values for deliverables

3. Put the DVD of SQL Server 2008 R2 Enterprise or Datacenter Edition into the local DVD drive and extract integritycheck_SQL2008R2.zip to the folder that contains the FCIV tool.
4. Open a command prompt and change to the directory to which the integritycheck_SQL2008R2.zip has been extracted.
5. Execute “integritycheck_sqlserver2008R2.cmd” and verify that the feedback matches the following picture:



```
Administrator: Command Prompt
D:\integritycheck_SQL2008R2>integritycheck_sqlserver2008R2.cmd f:
    The verification process may take several minutes...
    The files on the DVD are being scanned to validate their integrity
    .....
All files verified successfully
The DVD in drive f: is an authentic SQL Server2008 R2
D:\integritycheck_SQL2008R2>_
```

Figure 1: Successful verification of integrity

3.3.2 Installing the product

The description in this chapter focuses on a typical way of installing the database engine of SQL Server 2008 R2. For a more general overview over all options for the SQL Server setup please refer to [AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10sq_GetStart/html/6ad23de1-2bab-4933-9122-c09f5565028d.htm].

Please note that the installation procedure presented in this chapter belongs to the Enterprise Version of SQL Server 2008 R2. The installation procedure for the Datacenter Edition follows the same structure.

The SQL Server Installation Wizard is Windows Installer-based. It provides a single feature tree for installation of all SQL Server components.

To install SQL Server 2008 R2 one has to insert the SQL Server installation media and double-click setup.exe in the root folder. This installer will by default install the version of SQL Server that fits to the installed Operating System (x64).

For local installations, Setup has to run as an administrator.

For the case that the .NET Framework or the Windows installer (also referred to as "Hotfix for Windows (KB942288)") that are required (See also chapter 3.2) are not installed, SQL Server setup will offer their installation.

When the prerequisites are installed, the Installation Wizard will run the SQL Server Installation Center as seen in the following figure. To create a new installation of SQL Server 2008 R2, click New SQL Server stand-alone installation or add features to an existing installation".



Figure 2: Installing SQL Server 2008 R2 (I)

Next the System Configuration Checker (see Figure 3) will run a discovery operation on your computer. Setup log files have been created for the installation. For more information, see [AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10sq_GetStart/html/9d77af64-9084-4375-908a-d90f99535062.htm].

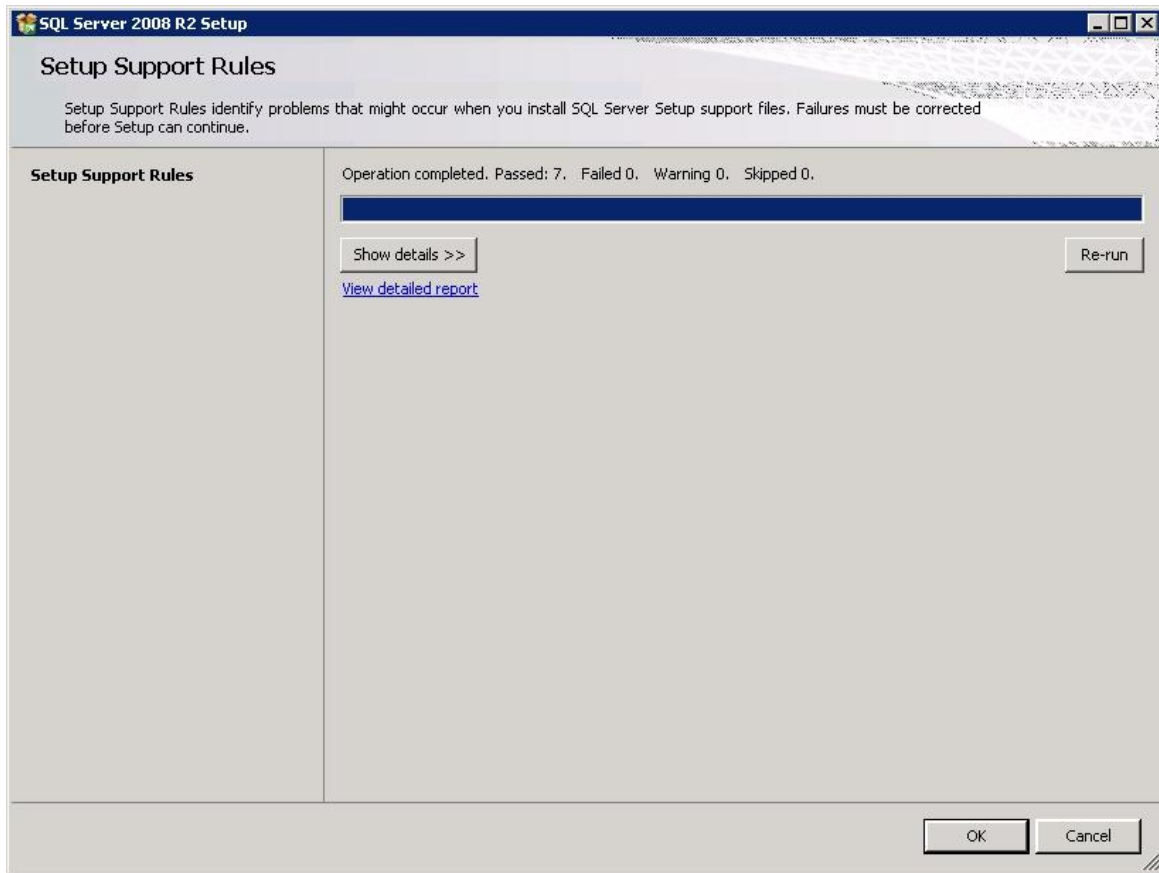


Figure 3: Installing SQL Server 2008 R2 (II)

On the Product Key page (see Figure 4), one selects a radio button to indicate whether installing a free edition of SQL Server, or a production version of the product that has a PID key. As only the Enterprise Edition of SQL Server has been certified one has to use a corresponding product key here. For more information, see [AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10sq_GetStart/html/e5186f02-dd91-47d0-8fa4-de3f41c76903.htm]

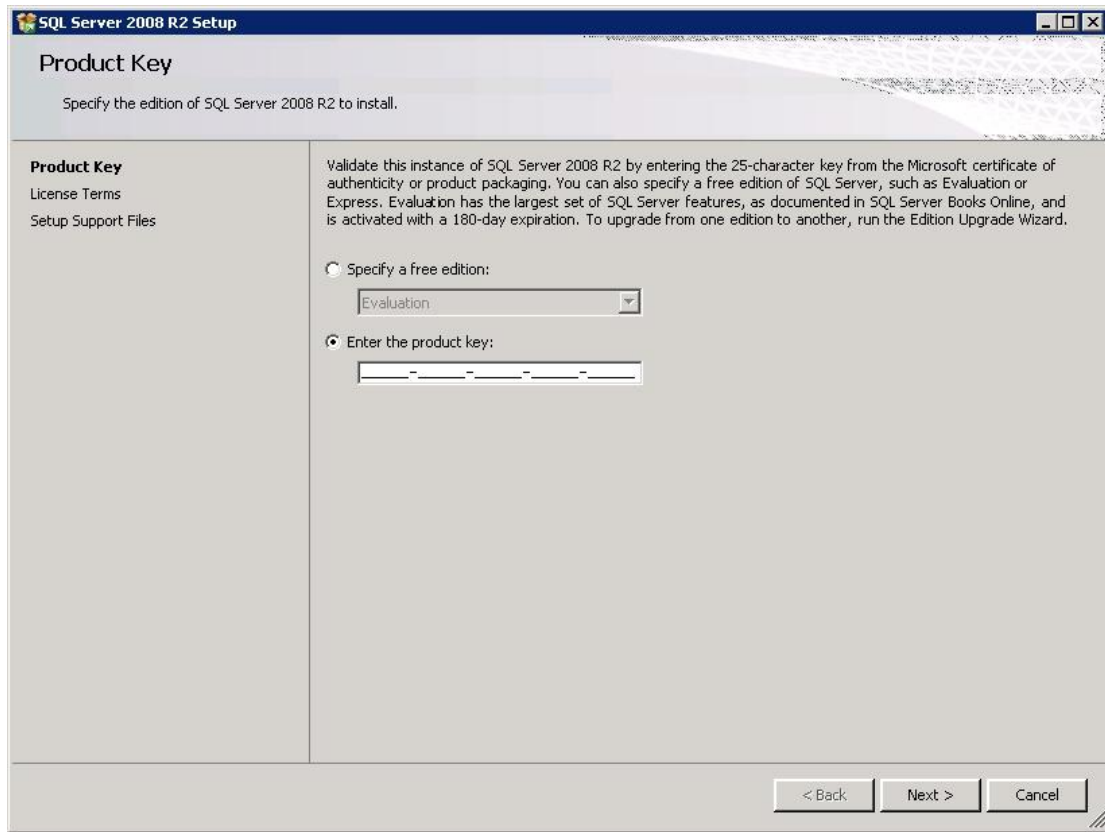


Figure 4: Installing SQL Server 2008 R2 (III)

On the License Terms page (Figure 5), one shall read the license agreement, and then select the check box to accept the licensing terms and conditions.

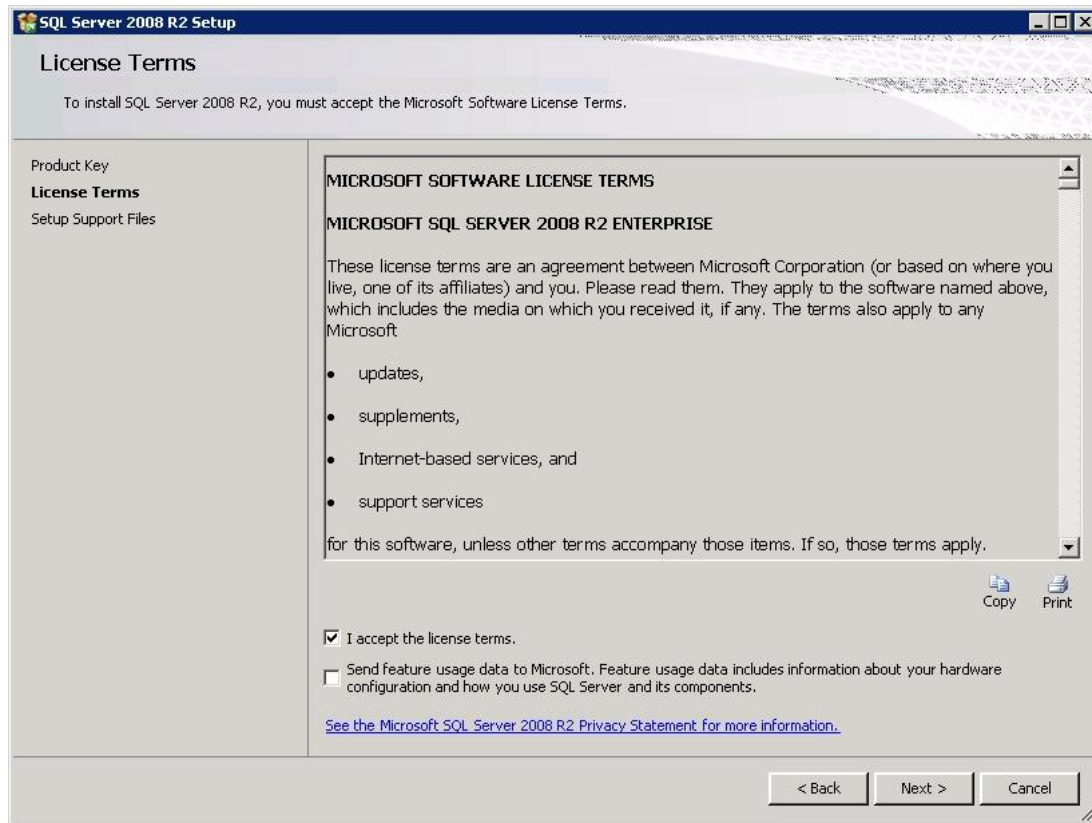


Figure 5: Installing SQL Server 2008 R2 (IV)

If all the other prerequisites have already been installed the Installation Wizard will then only copy the Setup Files to the hard disk as shown in Figure 6.

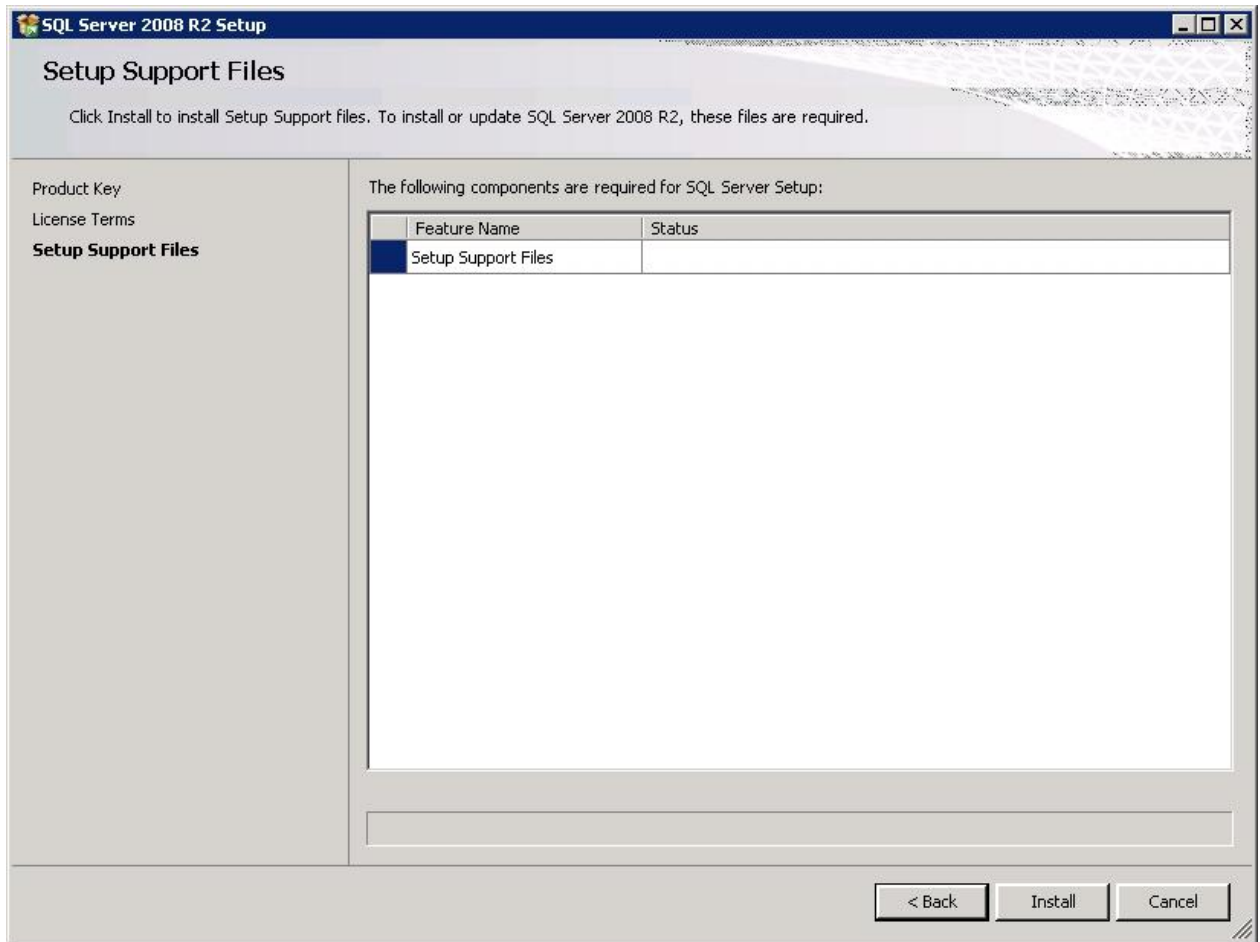


Figure 6: Installing SQL Server 2008 R2 (V)

The System Configuration Checker will verify the system state of the machine before Setup continues (Figure 7). Warning messages shown by the Configuration Checker (e.g. the one to see in Figure 7 that is shown because the Windows Firewall is active) shall be carefully considered but do not prevent the further installation.

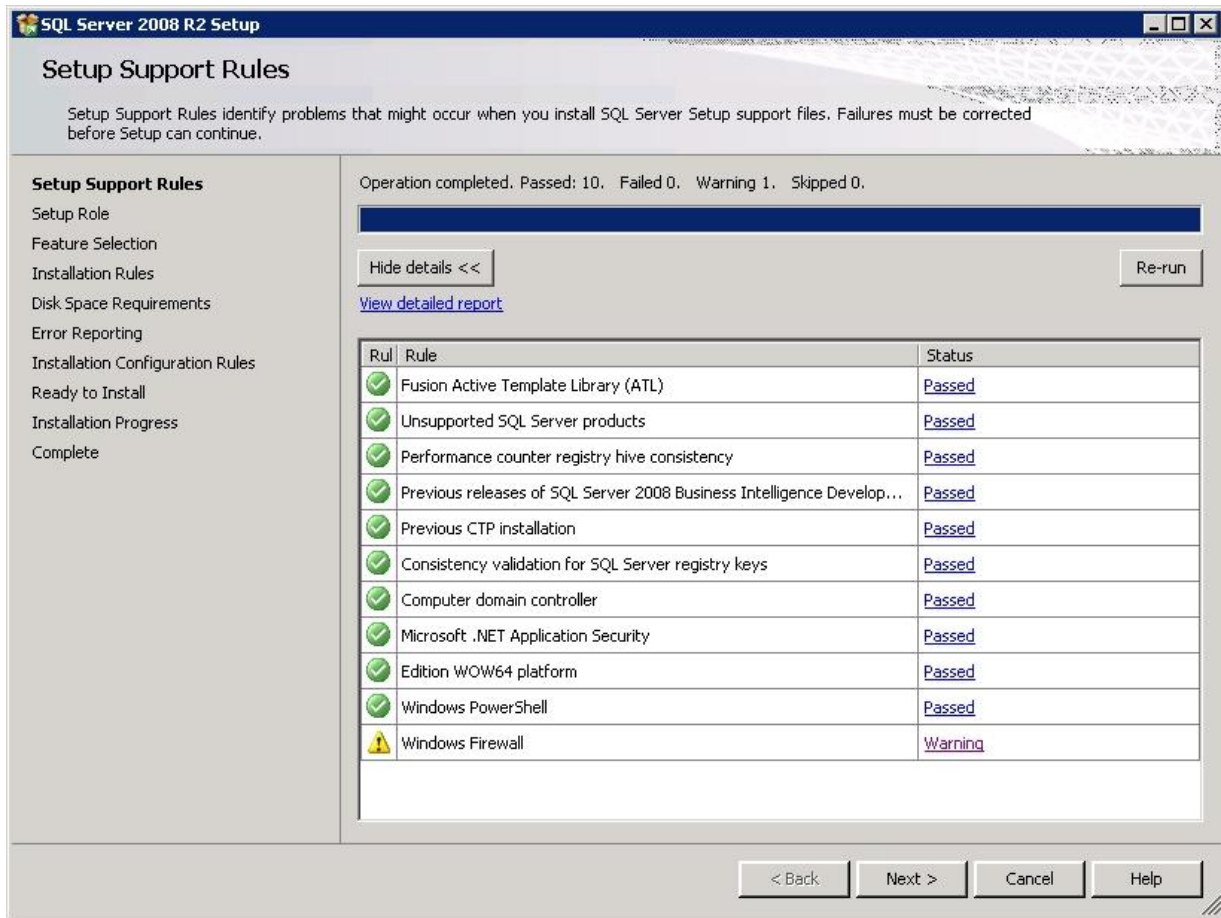


Figure 7: Installing SQL Server 2008 R2 (VI)

Next, the setup process will ask for the Setup Role. To install a local SQL Server one chooses “SQL Server Feature Installation”.

On the Feature Selection page, one can select the components for installation. A description for each component group appears in the right-hand pane after selecting the feature name. One can select any combination of check boxes.

For the certified version of the database engine of SQL Server 2008 R2 the following selection of components is recommended. It will install an instance of the database engine, tools for management and the documentation in form of SQL Server Books Online. According to an assumption of the evaluation process other components may only be installed if they are indispensable for the operation of the database engine.

One can also specify a custom directory for shared components by using the field at the bottom of the Feature Selection page.

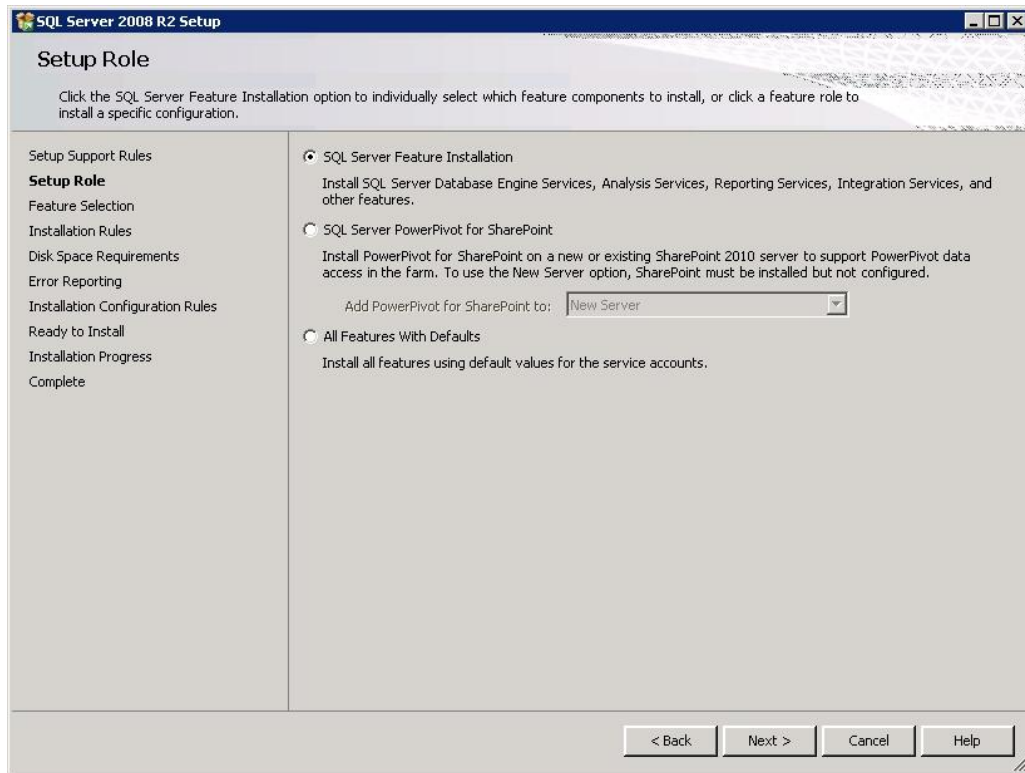


Figure 8: Installing SQL Server 2008 R2 (VII)

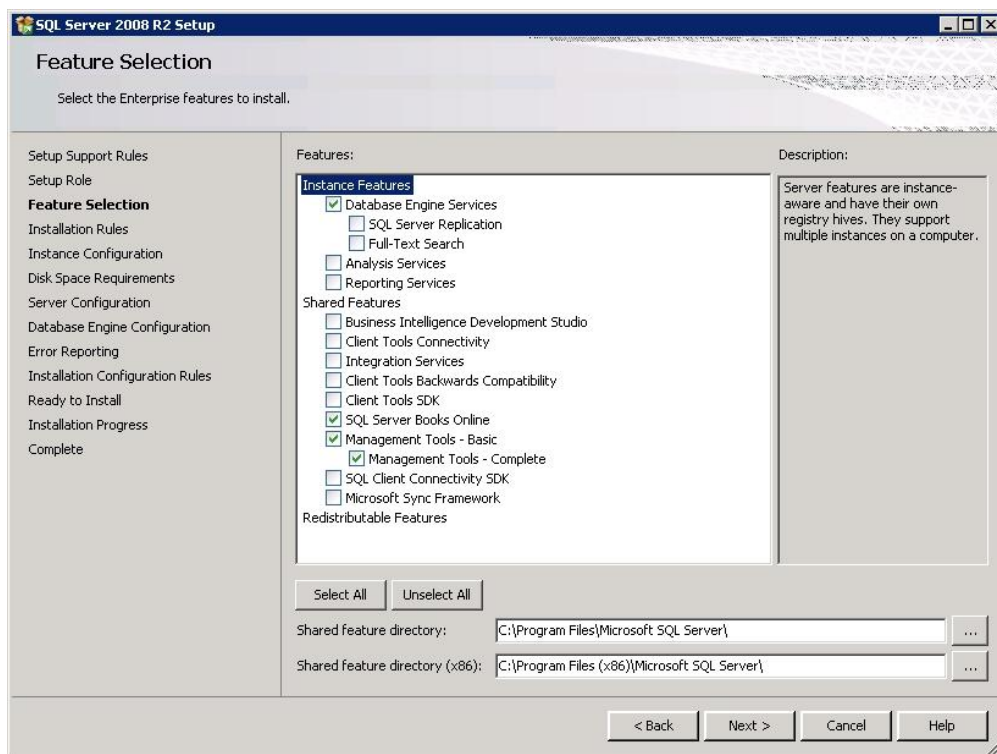


Figure 9: Installing SQL Server 2008 R2 (VIII)

Next, the System Configuration Checker will run one more set of rules to validate your computer configuration with the SQL Server features that have been selected.

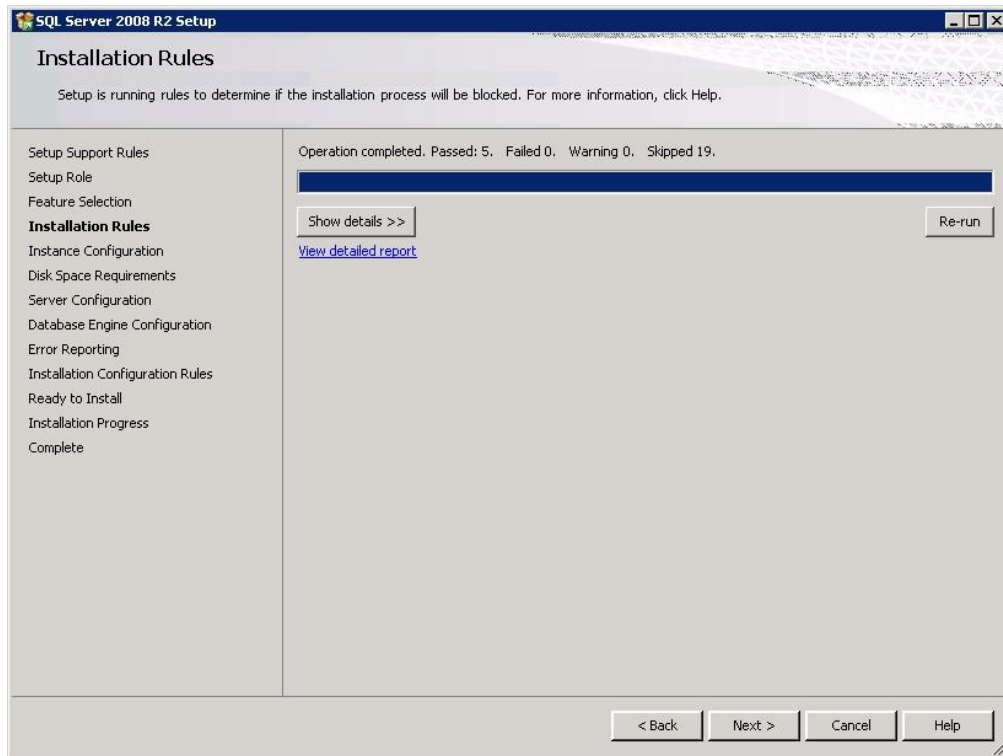


Figure 10: Installing SQL Server 2008 R2 (IX)

On the Instance Configuration page (see Figure 11), one shall specify whether to install a default instance or a named instance. For more information, see [AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10sq_GetStart/html/5bf822fc-6dec-4806-a153-e200af28e9a5.htm]

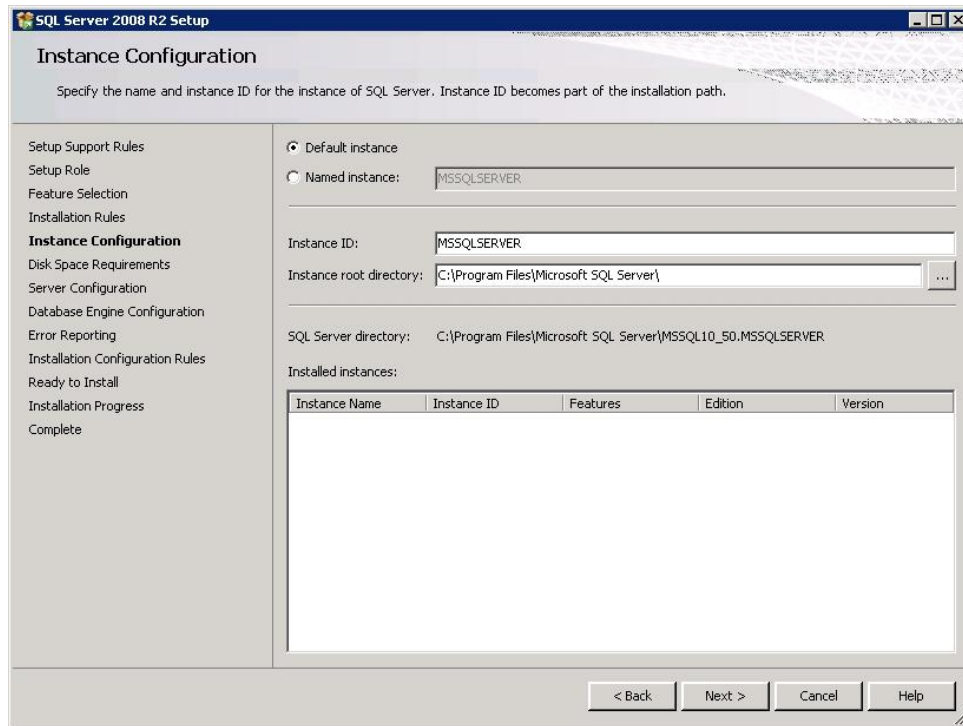


Figure 11: Installing SQL Server 2008 R2 (X)

The Disk Space Requirements page (see Figure 12) calculates the required disk space for the features you specify. It then compares the required space to the available disk space. For more information, see [AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10sq_GetStart/html/b9dc1b0a-1717-4e3e-b3d7-69397131c77a.htm]

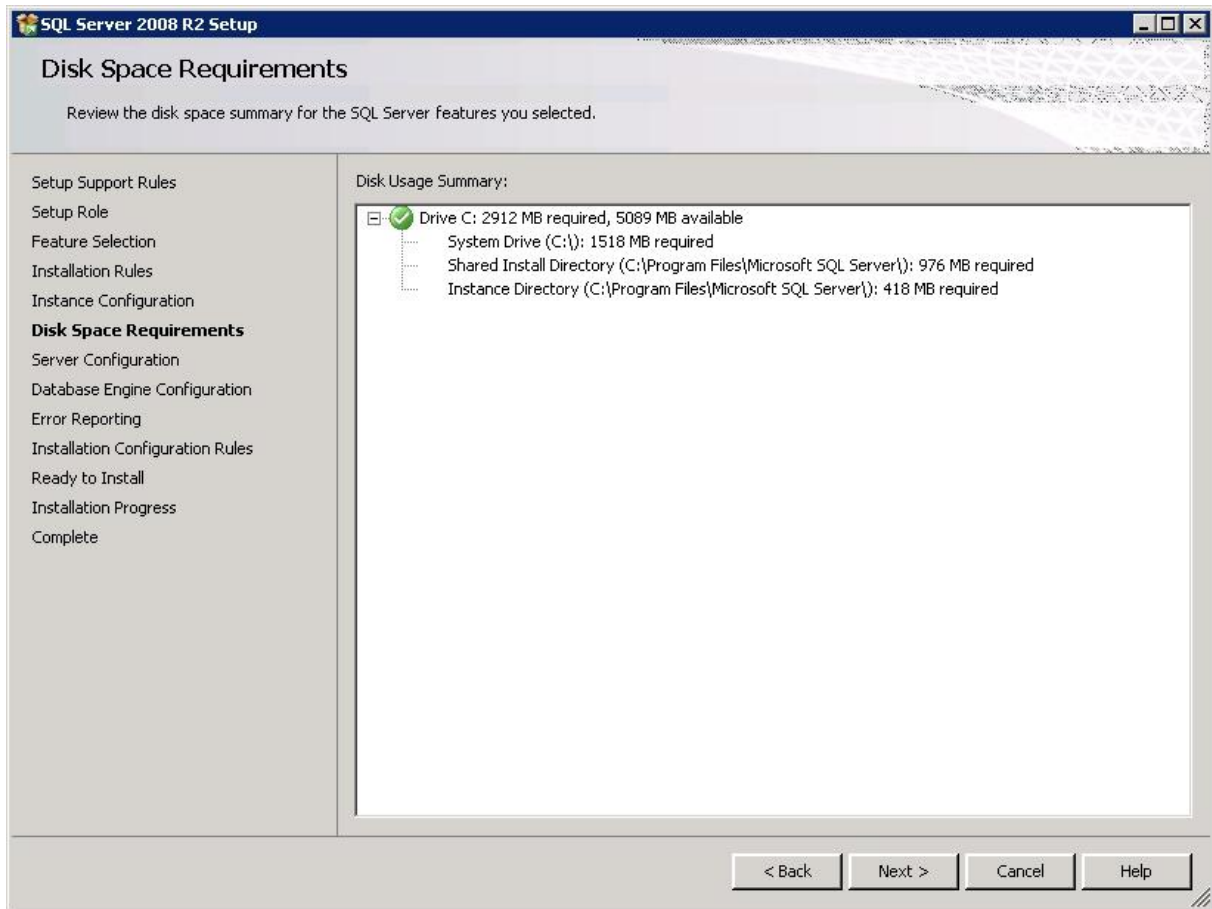


Figure 12: Installing SQL Server 2008 R2 (XI)

On the Server Configuration — Service Accounts page (see Figure 13), one specifies the login accounts for SQL Server services. The actual services that are configured on this page depend on the features that have been selected for installation.

One can assign the same login account to all SQL Server services, or one can configure each service account individually. One can also specify whether services start automatically, are started manually, or are disabled. It is recommended to configure service accounts individually to provide least privileges for each service, where SQL Server services are granted the minimum permissions they need to complete their tasks. In general, it is recommended not to use the service accounts that are created for SQL Server services for any other purposes.

The Server Configuration — Collation tab can be used to specify non-default collations for the Database Engine and Analysis Services. For more information, see [AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10sq_GetStart/html/e3986870-5be4-458b-b671-5ff12a27b022.htm]

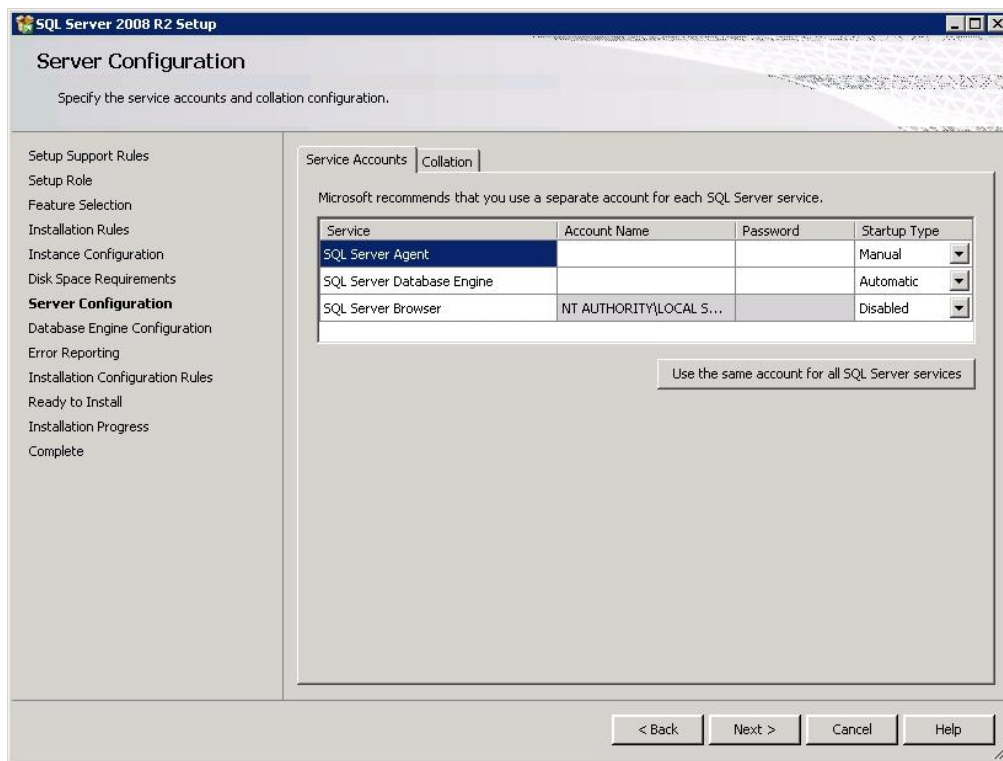


Figure 13: Installing SQL Server 2008 R2 (XII)

The Database Engine Configuration - Account Provisioning page (see Figure 14) can be used to specify the following:

- Security Mode — select Windows Authentication or Mixed Mode Authentication for the instance of SQL Server. If selecting Mixed Mode Authentication, one shall provide a strong password for the built-in SQL Server system administrator account. Please note that the SQL Server authentication will only be available if Mixed Mode authentication is chosen here.
- SQL Server Administrators — One must specify at least one system administrator for the instance of SQL Server. Adding the account under which SQL Server Setup is running can be done by clicking Add Current User. For more information, see [\[AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10sq_GetStart/html/834b26bc-49de-4033-88d5-6aa7b1609720.htm\]](http://ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10sq_GetStart/html/834b26bc-49de-4033-88d5-6aa7b1609720.htm)
- The Database Engine Configuration - Data Directories page (see Figure 14) can be used to specify non-default installation directories.

For more information, see [\[AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10sq_GetStart/html/9b1fa0fc-623b-479a-afc3-4f13bd850487.htm\]](http://ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10sq_GetStart/html/9b1fa0fc-623b-479a-afc3-4f13bd850487.htm)

The Database Engine Configuration - FILESTREAM page (see Figure 14) can be used to enable FILESTREAM for your instance of SQL Server. For more information, see [\[AGD;](#)

ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10sq_GetStart/html/641a10a1-ae52-4d26-8f1c-a032a4aeff02.htm

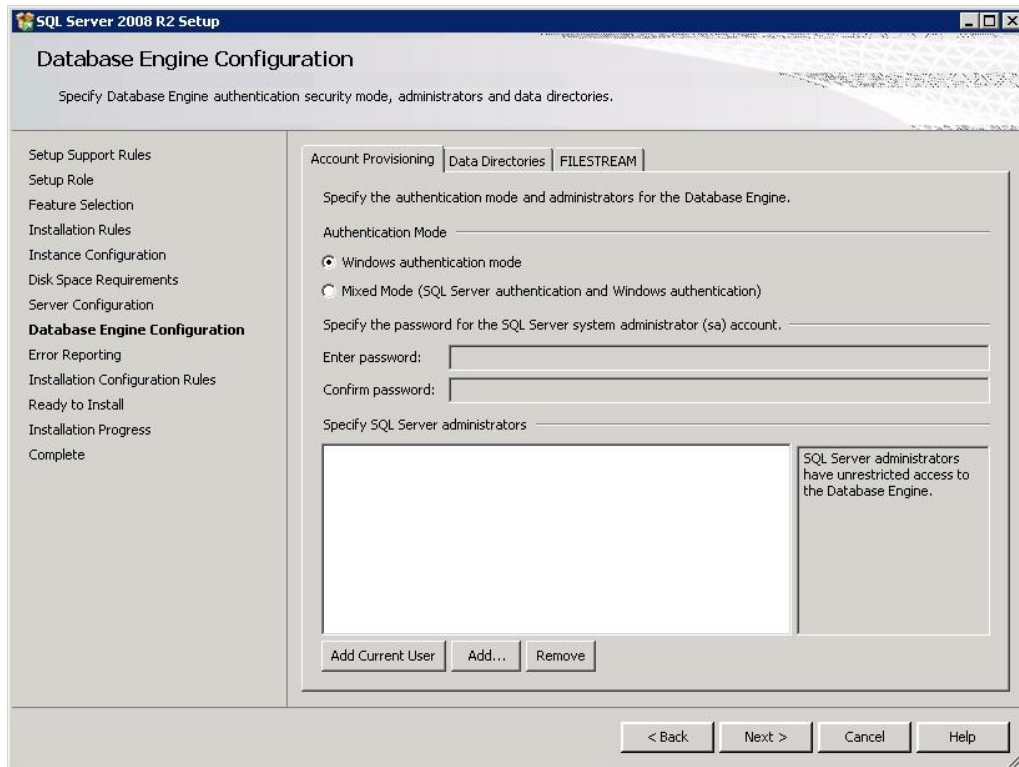


Figure 14: Installing SQL Server 2008 R2 (XIII)

On the Error and Usage Reporting page (see Figure 15), one specifies the information to be send to Microsoft that will help to improve SQL Server. By default, the option for error reporting is disabled. For more information, see [AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10sq_GetStart/html/e72c43b6-a2bd-4545-9aff-79c83b21180d.htm]

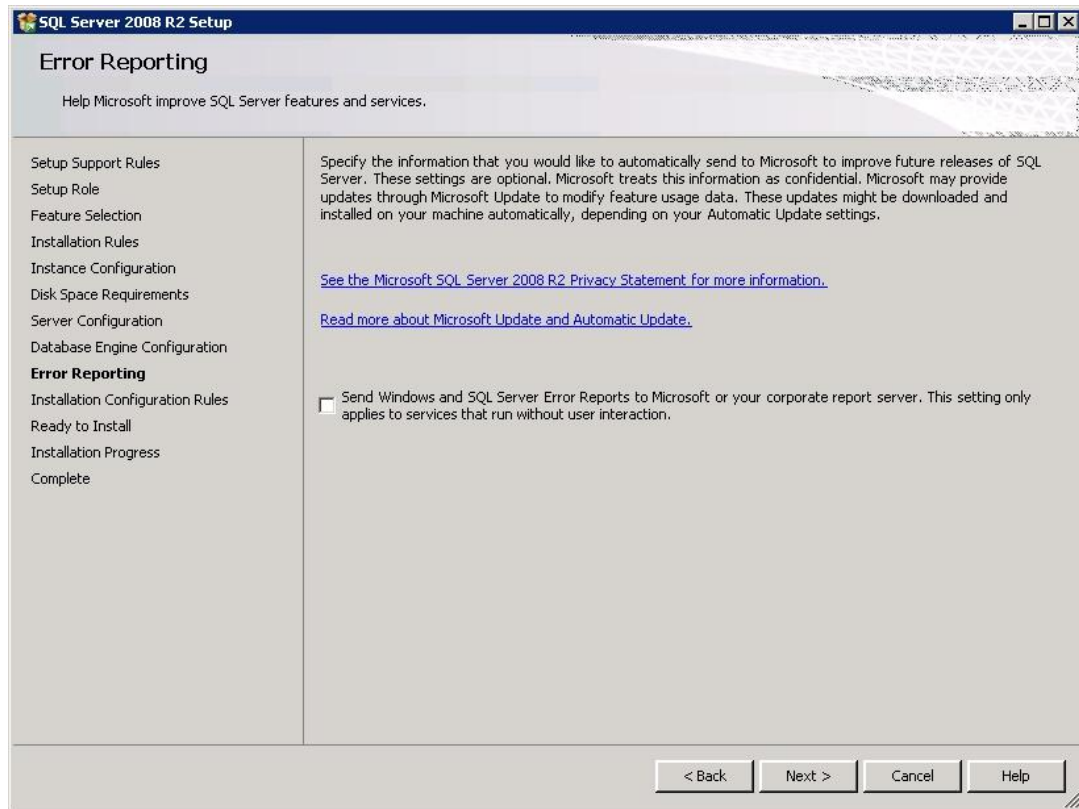


Figure 15: Installing SQL Server 2008 R2 (XIV)

The Ready to Install page (see Figure 16) shows a tree view of installation options that were specified during Setup.

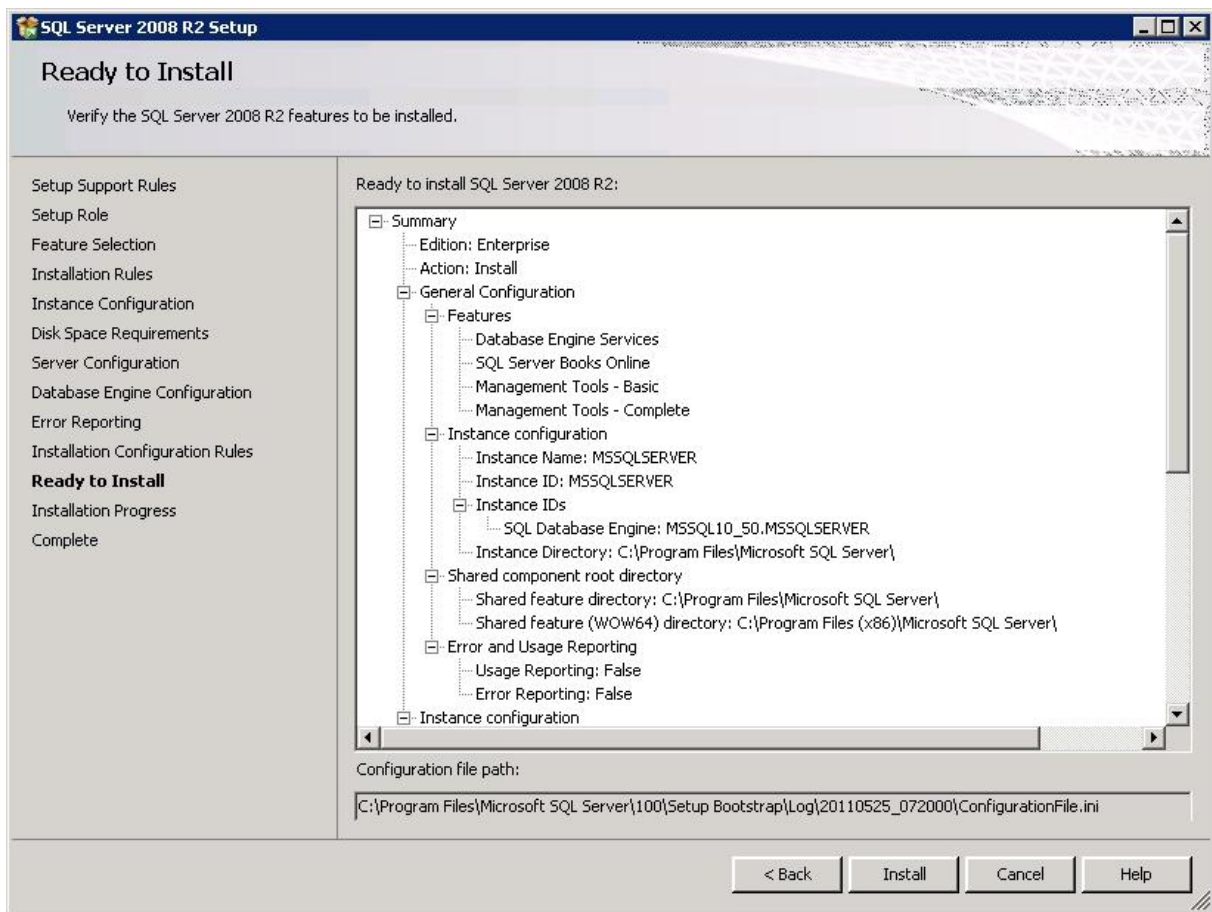


Figure 16: Installing SQL Server 2008 R2 (XV)

During installation, the Installation Progress page provides status so you can monitor installation progress as Setup proceeds.

After installation, the complete page (see Figure 17) provides a link to the summary log file for the installation and other important notes. The SQL Server installation process is finished after clicking Close.

If you are instructed to restart the computer, do so now. It is important to read the message from the Installation Wizard when you are done with Setup. For more information, see [AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10sq_GetStart/html/9d77af64-9084-4375-908a-d90f99535062.htm].

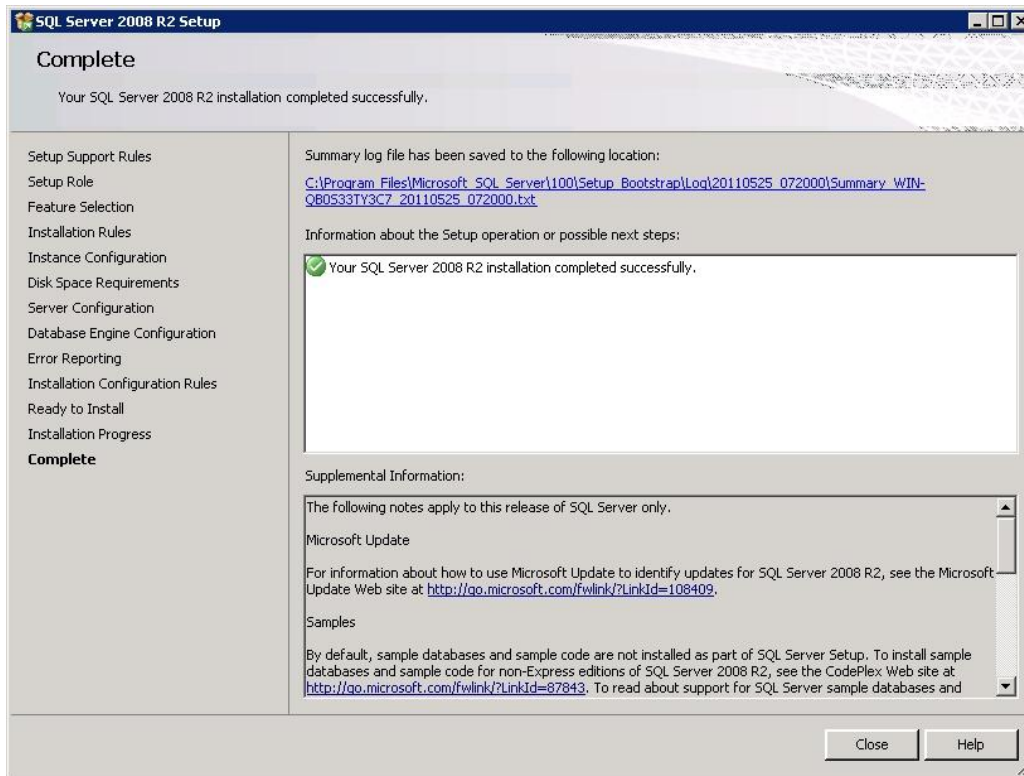


Figure 17: Installing SQL Server 2008 R2 (XVI)

3.3.3 Installing SP1

The Service Pack 1 (SP1) which is part of the evaluated version does not ship together with the product. It can be obtained via <http://www.microsoft.com/downloads/details.aspx?FamilyID=B9AA2DBA-7F20-4C0C-9AFD-1EEBEE5A94EA&displaylang=pt-br&displaylang=en>.

Before starting the installation process for SP1, the user shall verify the integrity of the installer file using the FCIV tool - which has already been used to verify the installation media - as follows:

- 1) Download the SP1 installer to a folder on disk.
- 2) Copy the FCIV tool into the same folder.
- 3) Open a command prompt and change to the folder.
- 4) Check the integrity of the download by executing the following command and verifying the return value:

Command	Expected return value
fciv SQLServer2008R2SP1-KB2528583-x64-ENU -sha1	896a448ef0274cc999955b0d5ef9a34ab51f0c3b sqlserver2008r2sp1-kb2528583-x64-enu.exe

Table 3: Commands to verify the integrity of SP1

The installation process for SP1 is self-explaining and does not require any settings specific to the evaluated version of the Database Engine of Microsoft SQL Server 2008 R2.

3.3.4 Checking the version of the product

After the installation process has been finished the admin shall finally determine whether the correct version of SQL Server 2008 R2 is installed. To do this he has to connect to the running database engine (using any T-SQL client) and execute the following command:

```
SELECT @@VERSION  
go
```

Using this command the TOE will return the name of the product platform (of which the TOE is the central part), the version number of the TOE and information about the Operating System.

For the certified version (which does not include the IA64 edition) the string that is returned in response to this command shall start with

- Microsoft SQL Server 2008 R2 (SP1) - 10.50.2500.0 (X64)

These strings include information on the concrete version of the SQL Server that has been installed (10.50.2500.0) and also show that the x64 edition has been installed.

3.3.5 Format of version numbers

Please note that due to reasons of backwards compatibility two different formats for the version numbers of SQL Server are available:

- The Product Version Number returned by the database engine (see also chapter 3.3.4) (10.50.2500.0)
- The Version Number of the executable files (2009.100.2500.0)

These version numbers are the same, though they will be in a different format and in fact the same number may be displayed in different formats.

The version number that is returned by the database engine using the select @@version statement as described in chapter 3.3.4 is labeled 10.50.xxxx.yy. The “xxxx” is the build number. For every new version that number is incremented. The “yy” is the number of rebuilds of the same build. On a few occasions, late in the development process if ever, the “yy” represents builds when another product needs to hard code a SQL Server build number before the final build.

For example, the 2nd rebuild of the 127th build of SQL Server is 10.50.0127.02. The rebuild would be 10.50.0127.03 and the next build would be 10.50.0128.00.

The historical fact is that the File Version Number (that is displayed by the Operating System after doing a right-click and choosing “Properties” on that file) was once a date and

it still has the year as it's first part. To retain backward compatibility with the software that uses these, the File Version Number format was not changed.

For that reason, also the „.100.“ in the File Version Number is equivalent to „.10.0.’ in the Product Version Number. Note also that leading and trailing zeroes in the Product Version Number are sometimes displayed.

3.3.6 Enabling the certified version

In the default installation of SQL Server 2008 R2 some of the Security Features that are important in the context of the evaluated version are not enabled.

Thus the administrator has to enable the Common Criteria Compliance option that enables:

- **Residual information protection:** This feature requires a memory allocation to be overwritten with a known pattern of bits before memory is reallocated to a new resource. Meeting the RIP standard can contribute to improved security; however, overwriting the memory allocation can slow performance. After the common criteria compliance enabled option is enabled, the overwriting occurs.
- **Login auditing** will be enabled. Each time a user successfully logs in to SQL Server, information about the last successful login time, the last unsuccessful login time, and the number of attempts between the last successful and current login times is made available. These login statistics can be viewed by querying the sys.dm_exec_sessions dynamic management view.
- After the common criteria compliance enabled option is enabled, a table-level **DENY** takes **precedence** over a column-level GRANT. When the option is not enabled, a column-level GRANT takes precedence over a table-level DENY.

To enable this option the administrator shall connect to the database engine and issue the following commands:

```
sp_configure 'show advanced options', 1;  
GO  
RECONFIGURE;  
GO  
sp_configure 'common criteria compliance enabled', 1;  
GO  
RECONFIGURE  
GO
```

These setting takes effect directly after the server has been restarted.

For more information please refer to [AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_4deptrbl/html/61766eea-c450-408d-af33-fbe7ef8c9ff2.htm]

3.3.7 Installing the logon triggers

The Security Function for Session Handling allows an administrator to restrict the ability of users to connect to the TOE based on

- The number of concurrent sessions per login
- User identity and the day of the week and time of the day

This functionality is implemented using the logon triggers of the TOE. (For more information about logon triggers please refer to [AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_4deptb1/html/2f0ebb2f-de10-482d-9806-1a5de5b312b8.htm])

This means that a trigger is executed every time a user is attempting to connect to the TOE. This trigger determines whether the user is allowed to establish a session at this time and denies session establishment if necessary.

The tables that store the information for this Security Function, the triggers and the Stored Procedures to manage this functionality have to be installed as they do not ship together with the database engine of SQL Server.

The installation can easily be done by executing the script "Install_cc_triggers.sql" that can be obtained via [WEB].

This script will install/create:

The tables:

- `dbo.denied_logins_A54E382458CA11DB8373B622A1EF5492`
This table contains the weekly intervals in which logins are not allowed to connect to SQL Server. The table should not be modified directly. The following stored procedures should be used instead:
 - `master.dbo.sp_deny_login`
 - `master.dbo.sp_revoke_login_denies`
- `dbo.maximum_number_of_connections_per_login_A54E382458CA11DB8373B622A1EF5492`
This table contains the value for the maximum number of connections per login. It should not be modified directly. Use the following stored procs instead:
 - `master.dbo.sp_set_maximum_number_of_connections_per_login`
 - `master.dbo.sp_remove_maximum_number_of_connections_limit`

The view:

- `dbo.denied_logins`
This view dumps the contents of the table with the weekly intervals in human readable format.

The function

- `dbo.fn_is_original_login_denied_A54E382458CA11DB8373B622A1EF5492`
This function checks whether the original login (the one who created the session) is

allowed to logon at this time. EXECUTE permission for this function is granted to everyone.

The logon trigger

- trig_deny_access_A54E382458CA11DB8373B622A1EF5492
This trigger is executed on every LOGON attempt. It checks whether the login is allowed to logon at this time (based on the time of the day and the day of the week) and if NOT rejects the connection by raising an exception.
- trig_max_connections_A54E382458CA11DB8373B622A1EF5492
This trigger is executed on every LOGON attempt. It checks whether the login is allowed to logon at this time (based on the maximum number of concurrent session per user) and if NOT rejects the connection by raising an exception.

The Stored Procedures

- dbo.sp_deny_logon_internal_A54E382458CA11DB8373B622A1EF5492
This is an utility stored procedure and it is not supposed to be called directly
- dbo.sp_deny_logon (see chapter 5.4.1.8.1)
- dbo.sp_revoke_logon_denies (see chapter 5.4.1.8.2)
- dbo.sp_set_maximum_number_of_connections_per_login (see chapter 5.4.1.8.3)
- dbo.sp_remove_maximum_number_of_connections_limit (see chapter 5.4.1.8.4)
- sp_trace_setcategory (see chapter 5.4.1.6.1)
- sp_trace_setcategory_all (see chapter 5.4.1.6.2)

3.3.8 Setting up the trace process

According to [ST] the TOE has to be able to audit a minimum set of events. The TOE logs events in so called trace files. However this trace process is not automatically enabled but has to be created by the administrator.

This can be done by executing a T-SQL script named "EAL4_trace.sql", which can be downloaded from [WEB]. This script that shall be executed as 'sa' will install a trace process including all necessary events and ensure that this trace process is started every time the server starts.

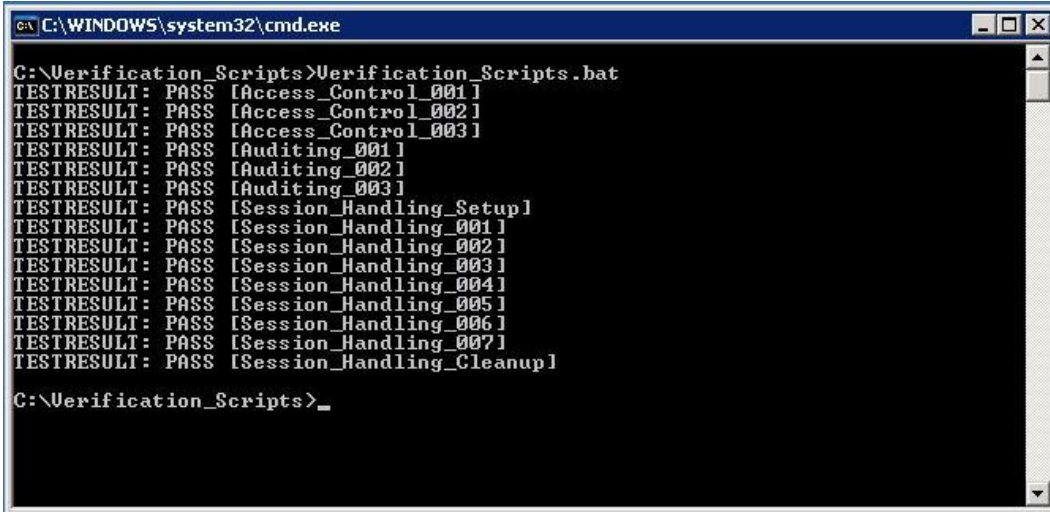
More information about the T-SQL commands which are used by this script can be found in chapter 5.4. More details about the events in this trace process and about trace in general can be found in chapter 6.

3.3.9 Basic verification of Security Functions

According to [PP] the administrator of SQL Server 2008 R2 shall be provided with a basic test to verify the correct operation of the Security Functions of the TOE.

This test is available in the file verification_script.zip that can be obtained via [WEB].

After unpacking the script locally (i.e. on the machine where the TOE is installed) it can be started by executing the file `Verification_Scripts.bat`. This file will execute a set of easy test cases to verify the operation of the Security Functions and print the results to the screen. The following screenshot shows the output of the script for the case that all test cases passed.



```
C:\WINDOWS\system32\cmd.exe
C:\Verification_Scripts>Verification_Scripts.bat
TESTRESULT: PASS [Access_Control_001]
TESTRESULT: PASS [Access_Control_002]
TESTRESULT: PASS [Access_Control_003]
TESTRESULT: PASS [Auditing_001]
TESTRESULT: PASS [Auditing_002]
TESTRESULT: PASS [Auditing_003]
TESTRESULT: PASS [Session_Handling_Setup]
TESTRESULT: PASS [Session_Handling_001]
TESTRESULT: PASS [Session_Handling_002]
TESTRESULT: PASS [Session_Handling_003]
TESTRESULT: PASS [Session_Handling_004]
TESTRESULT: PASS [Session_Handling_005]
TESTRESULT: PASS [Session_Handling_006]
TESTRESULT: PASS [Session_Handling_007]
TESTRESULT: PASS [Session_Handling_Cleanup]
C:\Verification_Scripts>_
```

Figure 18: Basic verification results

More detailed results (e.g. in case of any error) can be found in the file `Verification_Scripts_Result.txt` that is created in the directory from which the test cases were started. However, it should be noted that the file `Verification_Scripts_Result.txt` will contain error messages also for the case that all test cases passed as some test cases produce and expect errors of the database engine. An analysis of the content of the `Verification_Scripts_Result.txt` should only be necessary if one or more test cases have failed. In this case the content of the file has to be read in the context of the structure of the test scripts.

Please note that the current user has to have administrative privileges and Mixed Mode Authentication has to be enabled in order to run the test scripts.

4 SQL SERVER BOOKS ONLINE

The TOE is the security relevant part of a database management system, which primary purpose is to store and retrieve user data in a secure way.

Thus it is impossible to define, who the user of the TOE will be in practice. Many scenarios for the use of a database management system are possible. E.g.

- A user, who uses a T-SQL client for interaction with the database engine of SQL Server 2008 R2
- An application using the database engine of SQL Server 2008 R2

Books Online ([AGD]) provides all kinds of users with the necessary information, how the database engine of SQL Server 2008 R2 can be used.

The following links can be used as entry points into Books Online

Topic	Reference
Planning and architecture	[AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_0evalplan/html/f3363144-402d-49b6-b97e-a7b7e35e61a9.htm]
Development	[AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_1devconc/html/d9efe145-3306-4d61-bd77-e2af43e19c34.htm]
Querying and changing data	[AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_1devconc/html/1e57eb1f-e645-405c-b5e1-1cd6e2f62ec6.htm]
Deployment	[AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_4deptrbl/html/635e631c-74e9-4681-b5a9-4de985774a26.htm]
Operations	[AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_4deptrbl/html/377cb4cc-af31-4e36-8925-00bedb35f428.htm]
Security and Protection	[AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_4deptrbl/html/dfb39d16-722a-4734-94bb-98e61e014ee7.htm]
Technical Reference	[AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_5techref/html/e9533f6b-c48a-4f53-a7a0-379e092bb667.htm]

Table 4: Entry Points into Books Online

The following chapters are going to introduce the aspects for the secure administration and usage of SQL Server 2008 R2, which are specific to the certified version.

5 GUIDANCE ADDENDUM

This chapter contains the guidance addendum for the secure administration and usage of the TOE. It only covers the aspects of guidance, which are specific to the certified version of the database engine of SQL Server 2008 R2. It should be seen as a supplement to [AGD].

5.1 SQL Server startup flags

In its default configuration the process of the SQL Server database engine is running as a service under Windows 2008 Server R2 and automatically started after the start of the Operating System.

However in some situations it can be useful to start the engine using the "sqlservr.exe" directly and using certain modes of operation.

The following table lists the available options to be used with the "sqlservr.exe" that result in a certain mode of operation:

Option	Description
-c	Shortens startup time when starting SQL Server from the command prompt. Typically, the SQL Server Database Engine starts as a service by calling the Service Control Manager. Because the SQL Server Database Engine does not start as a service when starting from the command prompt, use -c to skip this step.
-f	Starts an instance of SQL Server with minimal configuration. This is useful if the setting of a configuration value (for example, over-committing memory) has prevented the server from starting.
-g	Specifies an integer number of megabytes (MB) of memory that SQL Server will leave available for memory allocations within the SQL Server process, but outside the SQL Server memory pool. The memory outside of the memory pool is the area used by SQL Server for loading items such as extended procedure .dll files, the OLE DB providers referenced by distributed queries, and automation objects referenced in Transact-SQL statements. The default is 256 MB.
-h	Reserves virtual address space for Hot Add memory metadata when AWE (Address Windowing Extension) is enabled with 32-bit SQL Server 2008. Required for Hot-Add memory with 32-bit AWE, but consumes about 500 MB of virtual address space and makes memory tuning more difficult. Not required for 64-bit SQL Server. Hot Add Memory is only available for Windows Server 2003, Enterprise and Datacenter editions. It also requires special hardware support from the hardware vendor.
-m	Starts an instance of SQL Server in single-user mode. When you start an

Option	Description
	instance of SQL Server in single-user mode, only a single user can connect, and the CHECKPOINT process is not started. CHECKPOINT guarantees that completed transactions are regularly written from the disk cache to the database device. (Typically, this option is used if you experience problems with system databases that should be repaired.) Enables the sp_configure allow updates option. By default, allow updates is disabled.
-n	Does not use the Windows application log to record SQL Server events. If you start an instance of SQL Server with -n, we recommend that you also use the -e startup option. Otherwise, SQL Server events are not logged.
-s	Allows you to start a named instance of SQL Server 2008 R2. Without the -s parameter set, the default instance will try to start. You must switch to the appropriate BINN directory for the instance at a command prompt before starting sqlservr.exe. For example, if Instance1 were to use \mssql\$Instance1 for its binaries, the user must be in the \mssql\$Instance1\bin directory to start sqlservr.exe -s instance1.
-T trace#	Indicates that an instance of SQL Server should be started with a specified trace flag (trace#) in effect. Trace flags are used to start the server with nonstandard behavior. For more information, see [AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_6tsql/html/b971b540-1ac2-435b-b191-24399eb88265.htm]
-x	Disables the keeping of CPU time and cache-hit ratio statistics. Allows maximum performance.
-e	Increases the number of extents that are allocated for each file in a filegroup. This option may be helpful for data warehouse applications that have a limited number of users running index or data scans.

Table 5: Startup Options for "sqlservr.exe"

The following modes shall not be used within the scope of the certified version as aspects of one or more Security Function as defined in [ST] may be affected.

- -f shall not be used within the scope of the certified version as aspects of one or more Security Function as defined in [ST] may be affected.
- -m: It cannot be guaranteed that all Security Functions are working in single user mode. Thus this mode must not be used within the certified version.
- -h This option is only available for Windows Server 2003, Enterprise and Datacenter editions and as such not supported by the certified version of SQL Server.

The following modes will require special care of the administrator. It is highly recommended not to use these modes within a productive environment within the scope of the certified

configuration. However it can be necessary to use these modes for debugging or maintenance purposes or within a specific environment:

- -n: Though the application log is not a direct part of any Security Function (Audit uses trace files) it is highly recommended not to use this mode within the certified configuration.
- -e is specific for data warehouse applications. It is highly recommended not to use this mode within the certified configuration.
- -T Trace#: Indicates that an instance of SQL Server should be started with a specified trace flag (trace#) in effect. Trace flags are used to start the server with nonstandard behavior. For more information, see [AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_6tsql/html/b971b540-1ac2-435b-b191-24399eb88265.htm].

The following modes will not affect the behavior of the database engine with respect to the Security Functions and can therefore be used in the scope of the certified version:

- -c: will only shorten the startup process of the engine but not affect the behavior of any Security Function
- -g: is an option for tuning the way memory is handled. No Security Function is affected by this mode. However, to ensure the correct operation of the database engine, this parameter shall not be used with values less than 64 MB.
- -s: Simply starts a further instance of the engine. The instances will work independently and enforce all Security Functions.
- -x: This mode can be used as the tuning which is done in this mode to allow maximum performance does not impact the Security Functions as defined in [ST].

Please note that the "sqlservr.exe" provides more options than listed in the previous table. However the other options do not represent a different mode of operation but would e.g. allow the administrator to specify another path for database files or error logs.

A complete overview of the options for "sqlservr.exe" can be found in [AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_4deptrbl/html/d373298b-f6cf-458a-849d-7083ecb54ef5.htm].

5.2 Administration Interface

The complete management functionality of the TOE as defined in the ST is available via the use of T-SQL commands or Stored Procedures which can be called using a T-SQL command. In this way any T-SQL conformant client can be used for administration.

The SQL Server Management Studio which ships together with the TOE comprises a T-SQL client in a comfortable GUI can be used for administration.

However the functionality of the GUI has not been evaluated.

To manage the services associated with SQL Server, to configure the network protocols used by SQL Server, and to manage the network connectivity configuration from SQL

Server client computers the SQL Server Configuration Manager tool can be used. The settings are stored and changed in the Operating System.

SQL Server Configuration Manager is a Microsoft Management Console snap-in that is available from the Start menu, or can be added to any other Microsoft Management Console display.

- SQL Server Configuration Manager can be used to start, pause, resume, or stop the services of SQL Server 2008 R2, to view service properties, or to change service properties.
- SQL Server 2008 R2 supports Shared Memory, TCP/IP, Named Pipes, and VIA protocols for its communication. These protocols can be managed (e.g. disabled and enabled) using SQL Configuration Manager. For information about choosing a network protocols see also [AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_4deptrbl/html/6565fb7d-b076-4447-be90-e10d0dec359a.htm] However the VIA protocol shall not be used within the certified version of the product (see also chapter 7).

More detailed information about the functionality which is provided by the SQL Server Configuration Manager can be found in [AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10sq_GetStart/html/e6beaea4-164c-4078-95ae-b9e28b0aefe8.htm].

5.3 User Interface

A user without administrative permissions can only connect to the TOE via the T-SQL interface using any T-SQL client via the protocols, which have been enabled by the administrator (see chapter 5.2).

The SQL Server Management Studio which ships together with the TOE comprises a T-SQL client, which can be used. (see also [AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_4deptrbl/html/f289e978-14ca-46ef-9e61-e1fe5fd593be.htm]). However the functionality of the GUI has not been evaluated.

For a complete overview over the T-SQL language please refer to the links under [AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_6tsql/html/dbba47d7-e08e-4435-b876-35dced1f325d.htm].

5.4 Security Functions relevant for administration and use of the TOE

The following chapters list the Security Functions of the TOE as defined in [ST] and describe, which parts of these Security Functions are accessible for administrators and users.

5.4.1 Security Management

As the name implies, the Security Function “Security Management (SF.SM)” as defined in [ST] is the core function for the secure management of the TOE.

For users without any administrative permission this Security Function does not have any accessible part.

For administrators this Security Functions comprises the following aspects:

- Add and delete logins on an instance level
- Add and delete users on a database level
- Add and delete group memberships (for database groups and server groups)
- Create and destroy database groups
- Create, Start and Stop Security Audit
- Include and exclude Auditable events
- Define the mode of authentication for every login
- Modify the action to take in case the audit file is full
- Manage Attributes for Session Establishment

The Security Function SF.SM comprises all aspects, which are relevant for the administration of the Security Functions Identification & Authentication (SF.I&A), Security Audit (SF.AU) and Session Handling (SF.SE). Only the Security Function Access Control (SF.AC) contains an additional aspect for administration: The possibility to grant and deny permissions to users (see chapter 5.4.3).

The following chapters introduce the commands which can be used via any T-SQL client to perform the operations mentioned before. More details about the commands can be found in [AGD]. Note: For some operations the following chapters list stored procedures to start the operation as well as T-SQL commands. For these cases the T-SQL commands shall be used primarily as the Stored Procedures are legacy commands and will be removed in a future release.

5.4.1.1 Add and delete logins on an instance level

To add and delete logins on an instance level the following T-SQL commands can be used. These commands are also used to specify the type of the login (whether it is associated with a Windows user account or a SQL login) as one has to decide about the type of the login during creation time. Please note that SQL logins are only available if the Mixed Mode Authentication has been chosen during the installation process.

Command	Purpose	Reference in [AGD]
sp_addlogin	Add a login	[AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_6tsql/html/030f19c3-a5e3-4b53-bfc4-de4bfca0fddc.htm]
Create Login	Add a login	[AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_6tsql/html/eb737149-7c92-4552-946b-91085d8b1b01.htm]
sp_droplogin	Delete a login	[AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_6tsql/html/e58684d1-c394-48de-906e-da6ee91100c3.htm]
Drop Login	Delete a login	[AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_6tsql/html/acb5c3dc-7aa2-49f6-9330-573227ba9b1a.htm]

Table 6: Commands to add and delete logins**5.4.1.2 Add and delete users on a database level**

To add or delete users from/to a database the following commands can be used:

Command	Purpose	Reference in [AGD]
Sp_adduser	Add user	[AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_6tsql/html/61a40eb4-573f-460c-9164-bd1bbfaf8b25.htm]
Create user	Add user	[AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_6tsql/html/01de7476-4b25-4d58-85b7-1118fe64aa80.htm]
Sp_dropuser	Delete user	[AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_6tsql/html/e28f18f9-7ecf-4568-89f4-fe5c520df386.htm]
Drop user	Delete user	[AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_6tsql/html/d6e0e21a-7568-4321-b6d6-bcfba183a719.htm]

Table 7: Commands to add and delete users**5.4.1.3 Add and delete group memberships**

To add or delete users from/to a database role/group or a server scoped group the following commands can be used:

Command	Purpose	Reference in [AGD]
sp_addrolemember	Add a database user to a group	[AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_6tsql/html/a583c087-bdb3-46d2-b9e5-3921b3e6d10b.htm]
sp_addsrvrolemember	Adds a login to a server scoped group	[AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_6tsql/html/777f0e09-8ee5-4cb2-a3ac-939d02c3cd22.htm]
sp_droprolemember	Remove a database user from a group	[AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_6tsql/html/c2f19ab1-e742-4d56-ba8e-8ffd40cf4925.htm]
sp_dropsrvrolemember	Remove a login from a server scoped role	[AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_6tsql/html/7be99181-d221-49d0-9cb2-c930d8c044a0.htm]

Table 8: Commands to add and delete users from database and server groups

An overview over the predefined server roles that ship together with the product and their permissions can be found in [AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_4deptrbl/html/13d47a53-1b5a-466f-8117-d060aa8d943e.htm].

5.4.1.4 Create and delete database groups

The following commands can be used to create and delete database scoped groups.

Command	Purpose	Reference in [AGD]
Sp_addrole	Add a group	[AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_6tsql/html/e8a21642-8440-419a-8585-93d3d9d44f00.htm]
Create role	Add a group	[AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_6tsql/html/b0cd54ad-e81d-4d71-acec-8a6d7261ca08.htm]
Sp_droprole	Delete a group	[ADG, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_6tsql/html/889ee074-00f8-40a9-bddb-d7d3ef0cbc19.htm]
Drop Role	Delete a group	[AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_6tsql/html/1f6f13ae-56a2-4ef1-93f5-8e6151b83e1d.htm]

Table 9: Commands to create and destroy database groups

An overview over the predefined database roles that ship together with the product and their permissions can be found in [AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_4deptrbl/html/a08108a3-f1fb-43ac-a264-3f2f9749db5d.htm].

5.4.1.5 Create, Start and Stop Security Audit

The following commands can be used to create, start and stop a trace process. When creating a new trace process one has to specify, what should happen in the case where the audit file is full.

Command	Purpose	Reference in [AGD]
Sp_trace_create	Create a new trace process	[AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_6tsql/html/f3a43597-4c5a-4520-bcab-becdbbf81d2e.htm]
Sp_trace_setstatus	Start and Stop a trace process	[AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_6tsql/html/29e7a7d7-b9c1-414a-968a-fc247769750d.htm]

Table 10: Commands to create, start and stop audit

Please note that a newly created trace will be in a stopped state until it is started using sp_trace_setstatus.

5.4.1.6 Include and exclude Auditable events

The following commands can be used to include and exclude auditable events from/to a trace file and to apply a filter to a trace.

Command	Purpose	Reference in [AGD]
Sp_trace_setevent	Include and exclude auditable events	[AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_6tsql/html/7662d1d9-6d0f-443a-b011-c901a8b77a44.htm]
Sp_trace_setfilter	Apply a filter to a trace	[AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_6tsql/html/11e7c7ac-a581-4a64-bb15-9272d5c1f7ac.htm]

Table 11: Commands to include and exclude auditable event

Please note that a trace process has to be in a stopped state before a filter can be applied and has to be started over after the filter has been applied.

5.4.1.6.1 Sp_trace_setcategory

This Stored Procedure allows the administrator to enable or disable a given data column for all events in a given trace category.

Syntax

```
sp_trace_setcategory [@traceid=] traceid
```

```
,[@categoryid=] categoryid  
,[@columnid=]columnid  
,[@on=] on
```

Arguments

[@traceid=] traceid

This is the ID of the trace process in which the events shall be enable/disabled. Traceid is of type INT.

[@categoryid=] categoryid

This is the ID of the category (sys.trace_categories) of which all events shall be enabled/disabled. Categoryid is of type INT.

[@columnid=]columnid

This is the ID of the column (sys.trace_columns) that shall be enabled/disabled for all events in the category. Columnid is of type INT.

[@on=] on

This argument specifies whether the events shall be enable (1) or disabled (0). On is of type BIT.

Return Values

0 (Success) or >0 (Failure)

Permissions

Requires the EXECUTE permission on the Stored Procedure and ALTER TRACE permission.

Remarks

Please not that other than standard system Stored Procedures that do live in the sys. – schema this Stored Procedure is stored in the dbo-schema of the master database.

5.4.1.6.2 Sp_trace_setcategory_all

This Stored Procedure allows the administrator to enable or disable all valid data column for all events in a given trace category.

Syntax

```
sp_trace_setcategory [@traceid=] traceid  
                      ,[@categoryid=] categoryid  
                      ,[@on=] on
```

Arguments

[@traceid=] traceid

This is the ID of the trace process in which the events shall be enable/disabled. Traceid is of type INT.

[@categoryid=] categoryid

This is the ID of the category (sys.trace_categories) of which all events shall be enabled/disabled. Categoryid is of type INT.

[@on=] on

This argument specifies whether the events shall be enable (1) or disabled (0). On is of type BIT.

Return Values

0 (Success) or >0 (Failure)

Permissions

Requires the EXECUTE permission on the Stored Procedure and ALTER TRACE permission.

Remarks

Please note that other than standard system Stored Procedures that do live in the sys. – schema this Stored Procedure is stored in the dbo-schema of the master database.

5.4.1.7 Define the mode of authentication for every login

The mode of authentication for every login of the TOE has to be determined at creation time. The administrator, who creates a new login has to specify, whether a Windows login should be created or a SQL login. This is done via the parameter WINDOWS of the CREATE LOGIN command (See Chapter 5.4.1.1).

5.4.1.8 Manage Attributes for Session Establishment

The following stored procedures can be used to manage the attributes for session establishment. After a default installation of the engine as described in chapter 3.3 of this document the maximum number of sessions per user is set to 5 and initially no further default deny rules are existing.

5.4.1.8.1 Sp_deny_logon

This Stored Procedure allows the administrator to deny session establishment to a certain login based on the day of the week and the time of the day.

Syntax

```
sp_deny_logon [@login_name=] 'login'
               ,[@start_weekday=] start_weekday
               , [ @start_time =] 'start_time'
               ,[@end_weekday=] end_weekday
               ,[@end_time=] 'end_time'
```

Arguments

[@login_name=] 'login'

Is the name of the login. 'login' is of data type **sysname**.

[@start_weekday=] start_weekday

Is the day of the week where the session deny should start. Start_weekday is **tinyint** according to the @@DATEFIRST setting (i.e. 1 means Sunday in the default setting for @@DATEFIRST).

[@start_time =] 'start_time'

Is the time of the day where the session deny should start. Start_time is of **nvarchar(12)**, in format hh:mm:ss.000 (the last three digits represent milliseconds)

[@end_weekday=] end_weekday

Is the day of the week where the session deny should end. end_weekday is **tinyint** according to the @@DATEFIRST setting (i.e. 1 means Sunday in the default setting for @@DATEFIRST).

[@end_time=] 'end_time'

Is the time of the day where the session deny should end. end_time is of **nvarchar(12)**, in format hh:mm:ss.000 (the last three digits represent milliseconds)

Return Values

0 (Success) or >0 (Failure)

Remarks

This Stored Procedure can be called with any @@datefirst setting and the start of the interval given can be > than the end of the interval. In this case it splits the passed interval into two intervals.

Please note that other than standard system Stored Procedures that do live in the sys. – schema this Stored Procedure is stored in the dbo-schema of the master database.

Permissions

Requires the CONTROL SERVER permission.

5.4.1.8.2 Sp_revoke_logon_denies

This Stored Procedure allows an administrator to revoke all denies from a certain login.

Syntax

sp_revoke_logon_denies [@login_name=]'login'

Arguments

[@login_name=] 'login'

Is the name of the login for which all denies shall be revoked. 'login' is of data type **sysname**.

Return Values

0 (Success) or >0 (Failure)

Remarks

Please note that other than standard system Stored Procedures that do live in the sys. – schema this Stored Procedure is stored in the dbo-schema of the master database.

Permissions

Requires the CONTROL SERVER permission.

5.4.1.8.3 Sp_set_maximum_number_of_connections_per_login

This Stored Procedure allows the administrator to set the maximum number of connections that are allowed per login. This value is a global value that is valid for all logins.

Syntax

```
dbo.sp_set_maximum_number_of_connections_per_login  
[@max_connections=] max_connections
```

Arguments

```
[@max_connections=] max_connections
```

New value for the maximum number of allowed connection per login. Max_connections is of data type INT.

Return Values

0 (Success) or >0 (Failure)

Remarks

Please note that other than standard system Stored Procedures that do live in the sys. – schema this Stored Procedure is stored in the dbo-schema of the master database.

Permissions

Requires the CONTROL SERVER permission.

5.4.1.8.4 Sp_remove_maximum_number_of_connections_limit

This Stored Procedure allows the administrator to remove the setting for the maximum number of connections that are allowed per login. After successfully executing this Stored Procedure the TOE will no longer enforce any limitation on the number of concurrent sessions per login.

Syntax

```
dbo.sp_remove_maximum_number_of_connections_limit
```

Arguments

-

Return Values

0 (Success) or >0 (Failure)

Remarks

Please note that other than standard system Stored Procedures that do live in the sys. – schema this Stored Procedure is stored in the dbo-schema of the master database.

Permissions

Requires the CONTROL SERVER permission.

5.4.2 Session Handling

The information about the

- last successful attempt to establish a session
- last unsuccessful attempt to establish a session
- number of unsuccessful login attempts since the last successful login

can be obtained via the dynamic management view `sys.dm_exec_sessions`.

SELECT permission on this management view is granted to public by default so that every user is able to retrieve the information from this view. The user will retrieve information about their current session plus the date and time of the last unsuccessful and successful login attempt (before the current session was established) and the number of unsuccessful login attempts since the last successful login. A user who has the VIEW SERVER STATE permission will see this information for all active sessions.

For more information about this view please refer to [AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_6tsql/html/2b7e8e0c-eea0-431e-819f-8ccd12ec8cfa.htm]

5.4.3 Access Control

The Security Function Access Control ensures that only users, which have appropriate permissions, are able to perform operations on objects, under the control of the Security Function. The complete description of the Security Function can be found in [ST].

For users without any administrative permission the Security Function Access Control is only accessible in so far that for every command, which is issued to the TOE, the Security Function will check, whether the user has the appropriate permissions.

A part of the Security Function is that it is possible for administrators to grant, revoke or deny permissions to users using the following commands:

Command	Purpose	Reference in [AGD]
Grant	Grant permission	[AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_6tsql/html/a760c16a-4d2d-43f2-be81-ae9315f38185.htm]
Revoke	Revoke permission	[AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_6tsql/html/9d31d3e7-0883-45cd-bf0e-f0361bbb0956.htm]
Deny	Deny permission	[AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_6tsql/html/9d31d3e7-0883-45cd-bf0e-f0361bbb0956.htm]

Table 12: Commands to grant, revoke and deny permissions

Based on the identity of the user, the group membership of the user and the granted or denied permissions the database engine will decide based on the following rules whether an operation that is requested by a user is allowed:

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

It should be noted that the permission check 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.

However, there are two cases for which the aforementioned rules are overridden:

1. A sysadmin, the owner of an object and owners of parent objects always have access
2. In the case of "Ownership Chaining" (see also [AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_4deptrbl/html/762249ee-881a-4c3e-b8c0-3a9475039aca.htm]) the access is allowed.

5.4.4 Identification & authentication

The Security Function Identification & Authentication ensures that each user has been successfully authenticated before any other operations on behalf of that user are allowed. The complete description of the Security Function can be found in [ST].

The Security Function Identification & Authentication only has a small user-accessible part, which is that every user will be authenticated when connecting to the TOE. This applies to administrators and to users without any administrative permissions.

The complete administrative part of this Security Function is covered by the Security Function Security Management (see chapter 5.4.1).

5.4.5 Security Audit

This Security Function ensures that the TOE produces audit logs for a set of security relevant actions. These audit logs are stored into trace files in the environment of the TOE. The complete description of the Security Function including the complete list of events can be found in [ST].

For the user of the database engine without any administrative permission the Security Function Security Audit does not have any user-accessible functionality.

The complete administrative part of this Security Function is covered by the Security Function Security Management (see chapter 5.4.1).

For further information about the trace functionality of the TOE please refer to chapter 6.

6 SQL Server Trace

The audit functionality of SQL Server 2008 R2 as defined in [ST] is realized by its trace functionality.

This chapter will provide information about the trace functionality, which are of specific relevance for the certification process of SQL Server 2008 R2.

Detailed information about the trace functionality in general can be found in [AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_4deptrbl/html/deb81e26-d55b-4973-ab83-6de3ca20971c.htm]

SQL Server has the possibility to maintain several trace processes in parallel and allows the authorized administrator to include and exclude a wide range of events to the processes. For each event a set of data columns can be included, which contains the detailed information about the event. An overview over all events and all columns which can be included into a trace process can be found in [AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_6tsql/html/7662d1d9-6d0f-443a-b011-c901a8b77a44.htm].

The definition of each trace process comprises:

- The events which are captured in the trace process
- The definition of filters which are applied before the events are captured
- The maximum size for the trace file
- The action to take in the case that the trace file is full
- The maximum number of trace files (in case the rollover option has been specified)

The following chapters introduce the information, which have to be audited according to [ST] and the events from the SQL Server 2008 R2 trace functionality, which can be used to trace these information.

6.1 Information to be audited

The following table lists all the events which need to be audited according to [ST]:

ID	Event	See Chapter
1	Start-up and shutdown of the audit functions	6.3.2
2	Start-up and shutdown of the DBMS	6.3.1
3	Use of special permissions (e.g., those often used by authorized administrators to circumvent access control policies)	6.3.3
4	All modifications to the audit configuration that occur while the audit collection functions are operating.	6.3.4
5	Successful requests to perform an operation on an object covered by the SFP	6.3.5
6	Unsuccessful revocation of security attributes for subjects and objects	6.3.6
7	Every use of the management functions: <ul style="list-style-type: none"> • Add and delete logins • Add and delete users • Add and delete group memberships (DB scoped groups, Server scoped groups) • Create and destroy database scoped groups • Create, Start and Stop Audit • Include and Exclude Auditable events • Define the mode of authentication • Define the action to take in case the audit file is full 	6.3.7
8	Modifications to the group of users that are part of a role.	6.3.7
9	Every use of the authentication mechanism including the final decision on authentication	6.3.8

Table 13: Events to be audited

For these events the following information need to be audited:

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

The following two chapters introduce the trace processes, which are used to audit all necessary events.

6.2 Role of the default trace

Every instance of the TOE runs a so called default trace process.

This process

- Is automatically started together with the TOE
- Logs a predefined set of events
- Can only be started and stopped using “sp_configure”
- It uses the rollover option using a maximum number of 5 files and 20 MB per file
- Is stored in a trace file named “log_x.trc” in the default log directory (usually \MSSQL\LOG).

More detailed information about the trace functionality of the TOE can be found in “Introducing SQL Trace” [AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_4deptrbl/html/deb81e26-d55b-4973-ab83-6de3ca20971c.htm]

The startup of the TOE (in form of the Audit Server Start and Stop event class) itself is usually only audited in this default trace of the TOE as no other trace processes are yet running in the early phase of startup. However, as the trace process as described in chapter 6.3 logs the start of the database engine in form of a user defined error message the default trace process is not mandatory for the certified version of SQL Server.

Please note that the default trace process cannot be started and stopped using “sp_trace_setstatus” but only via the use of the Stored Procedure sp_configure. Please see [AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_6tsql/html/d18b251d-b37a-4f5f-b50c-502d689594c8.htm] for more information.

For the information that has to be traced as described in chapter 6.1 it is recommended that the administrator of the TOE creates a separate trace process for the certified version of SQL Server 2008 R2, which includes all events which shall be audited according to [ST]. This allows the admin to pay special attention to all events which have to be audited according to [ST].

Such a “CC” Trace process is described in the following chapter.

6.3 The “CC Trace”

To trace all the events in the TOE, which are important according to [ST] it is recommended to create a separate trace process, which includes all necessary events as listed in Table 14 and all necessary data columns.

Such a trace process can easily be created using the script “EAL4_trace.sql” as described in chapter 3.3.8.

This script will create and start a trace process with all necessary events. This trace process uses the rollover option (having 100 files, 100 MB each) and the TOE will stop operation if any error occurs in this trace process.

The administrator is free to define other values for the number of trace files or the size per file. He is also free not to use the rollover option but to only have one trace file. However the option to stop the server in case an error occurs in the context of this trace process shall always be used.

In this context it is important to mention that the administrator should ensure that sufficient disc space is available for the trace files as the engine in its default configuration will stop operation if the trace process has to be stopped due to insufficient disc space.

For the case that the TOE stops operation due to insufficient disc space for the trace file the administrator should either provide additional disc space or backup and delete the “old” trace files before starting the engine again.

Per default the trace files for this process are written into the default log directory (usually \MSSQL\LOG) and named “cc_trace_TIMESTAMP³_x.trc”. The “CC Trace” process will start automatically after the TOE has been stopped and started again. However as in every other trace process, which uses the rollover option, a rollover will happen (i.e. a new trace file will be started) every time the trace process is started again.

If the script succeeds it will return a message including the internal ID of the trace process and information about the trace files, which are in use.

The following table lists all events, which are included in the “CC trace” process.

³ Please note that the timestamp, which is used as part of the filename for the trace files has a resolution of 1 second. Thus the execution of the script will abort with an error if the script is started twice in one second.

event #	Name	Information audited from Table 13
14	Audit Login	5, 3, 9
15	Audit Logout	5
18	Audit Server Starts and Stops	2
20	Audit Login Failed	5,9
42	Sp:starting	7
43	Sp:completed	7
102	Audit Database Scope GDR Event	5
103	Audit Schema Object GDR Event	5, 6
106	Audit Login Change Property Event	5
107	Audit Login Change Password Event	5
108	Audit Add Login to Server Role Event	5, 6, 7, 8
110	Audit Add Member to DB Role Event	5, 6, 7, 8
112	Audit App Role Change Password Event	5
114	Audit Schema Object Access Event	5
115	Audit Backup/Restore Event Audit DBCC Event	5
116	Audit DBCC Event	5
117	Audit Change Audit Event	5, 1
128	Audit Database Management Event	5
129	Audit Database Object Management Event	5
130	Audit Database Principal Management Event	5, 6, 7
131	Audit Schema Object Management Event	5
132	Audit Server Principal Impersonation Event	5
133	Audit Database Principal Impersonation Event	5
134	Audit Server Object Take Ownership Event	5
135	Audit Database Object Take Ownership Event	5
152	Audit Change Database Owner	5
153	Audit Schema Object Take Ownership Event	5
162	Audit User Error Message	7, 2
170	Audit Server Scope GDR Event	5, 6
171	Audit Server Object GDR Event	5, 6
172	Audit Database Object GDR Event	5, 6
173	Audit Server Operation Event	5

event #	Name	Information audited from Table 13
175	Audit Server Alter Trace Event	5
176	Audit Server Object Management Event	5
177	Audit Server Principal Management Event	5, 7, 8
178	Audit Database Operation Event	5
180	Audit Database Object Access Event	5

Table 14: Necessary audit events

The following chapters now introduce how these events of the SQL Server 2008 R2 trace mechanism can be used to audit all the information as required by [ST] and also provide information about relevant information, which are stored in every event.

6.3.1 Startup and shutdown of DBMS

The “Audit Server Starts” and “Stops” event class occurs when the Microsoft SQL Server service state is modified.

Please note that the startup of the TOE cannot be logged via this event as the trace process (other than the default trace) is not yet running when the TOE starts. The startup of the TOE is logged using the User Error Message as described in chapter 6.3.10 as early as possible (i.e. directly after the trace has been started). This event is fired by the script that also installs the CC trace process (see chapter 6.3).

The “Audit Server Starts” and “Stops” event contains the following attributes which are important in the context of this evaluation:

Attribute	Description
SPID	ID of the session on which the event occurred.
StartTime	Time at which the event started, if available.
LoginName	Name of the login of the user (either the SQL Server security login or the Microsoft Windows login credentials in the form of DOMAIN\username).
Success	1 = success. 0 = failure.
EventSubClass	Type of event subclass. 1=Shutdown, 2=Started, 3=Paused, 4=Continue

Table 15: Important attributes of “Audit Server Starts and Stops” event

Please note that the event which indicates that the server has been started will always show success as otherwise the server would not have been started.

A complete description of the event can be found in [AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_4deptrbl/html/8ddb55af-c77b-4d07-b803-a97320c0804e.htm]

6.3.2 Startup and shutdown of audit functions

The “Audit Change Audit event class” occurs whenever an audit trace modification is made. Modifications in the context of this event comprise specifically to stop and start a trace process.

This event contains the following attributes which are important in the context of this evaluation:

Attribute	Description
SPID	ID of the session on which the event occurred.
StartTime	Time at which the event started, if available.
LoginName	Name of the login of the user (either the SQL Server security login or the Microsoft Windows login credentials in the form of DOMAIN\username).
DBUserName	SQL Server database user name of the client.
SessionLoginName	The login name of the user who originated the session. For example, if you connect to SQL Server using Login1 and execute a statement as Login2, SessionLoginName displays Login1, while LoginName displays Login2. This data column displays both SQL Server and Windows logins.
Success	1 = success. 0 = failure. For example, a value of 1 indicates success of a permissions check and a value of 0 indicates failure of that check.
EventSubClass	Type of event subclass. 1=Audit started, 2=Audit stopped, 3=C2 mode ON, 4=C2 mode OFF

Table 16: Important attributes of “Audit Change Audit” event

More information about this event can be found in [AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_4deptrbl/html/8cfacc82-cee8-4199-a69e-acedecfc0b3b.htm]

Furthermore every trace file contains the “Trace Start” event as the first event and the “Trace Stop” event as the last event. However these events only show that the trace has been started or stopped and include no additional information beside the date and time of the event. Specifically the “Trace Stop” event is always considered to be successful (though it does not fill the data column Success).

6.3.3 Use of special permissions

Authorized administrators do not have to take a specific action before they are allowed to perform administrative actions. Hence the only event which can be audited for the use of these special permissions is the fact that an authorized administrator has logged on to the TOE. This is covered by the event as described in chapter 6.3.8.

6.3.4 Modifications to the audit configuration

The “Audit Server Alter Trace event class” occurs for all statements that check for the ALTER TRACE permission. Statements that check for ALTER TRACE include those used to create or configure a trace, or to set a filter on a trace.

This event contains the following attributes which are important in the context of this evaluation:

Attribute	Description
SPID	ID of the session on which the event occurred.
StartTime	Time at which the event started, if available.
LoginName	Name of the login of the user (either the SQL Server security login or the Microsoft Windows login credentials in the form of DOMAIN\username).
DBUserName	SQL Server database user name of the client.
SessionLoginName	The login name of the user who originated the session. For example, if you connect to SQL Server using Login1 and execute a statement as Login2, SessionLoginName displays Login1, while LoginName displays Login2. This data column displays both SQL Server and Windows logins.

Table 17: Important attributes of “Audit Server Alter Trace” event

It should be noted that it is not possible to modify the configuration of a trace process while this process is running. To apply a filter to a trace process the trace process has to be stopped first. Defining the setting, what should happen in case the audit files are full (ROLLOVER OPTION) is only possible at creation time. If this setting should be changed for a running trace process, the trace process would have to be stopped and a new trace process has to be created.

If it is necessary to stop one or both of the trace processes mentioned before while the TOE is still running (e.g. to change the configuration of the trace process) it should be considered to create a new trace process which contains all the relevant events and to start this new process before the CC trace process is stopped. In this way it can be ensured that the admin misses no important event.

6.3.5 Requests on operation

The [ST] requires that every successful request by a user to perform an operation on an object has to be audited by the TOE.

The sum of the events as listed in Table 14 is suitable to meet this requirement as all operations on objects, which can be performed, are covered by the set of these events.

It should be noted that most of the audit events in Table 14 result directly out of the access control functions of the TOE. This also means that if a user operation requires more than one permission the trace file will contain more than one audit event for this operation.

For example: If a user attempts to create a new table the CREATE TABLE permission and the ALTER SCHEMA permission on the corresponding schema are needed. Thus if a user attempts to create a table two events will show up in the trace file and the access control check for the operation can only be considered successful if both events show success.

6.3.6 Unsuccessful revocation

6.3.6.1 For objects

For objects within the TOE the only (implicitly) defined attributes which can be revoked are the corresponding Access Control Entries (ACE) which are used for access control.

By the use of the T-SQL commands REVOKE, DENY and GRANT authorized users are able to modify these ACEs. The use of the commands REVOKE and DENY can be seen as revocation in terms of [CC].

The following events are fired for every REVOKE, DENY and GRANT statement for specific objects:

- Audit Database Scope GDR Event
- Audit Schema Object GDR Event
- Audit Server Scope GDR Event
- Audit Server Object GDR Event
- Audit Database Object GDR Event

The sum of these events covers all REVOKE, DENY and GRANT statements, which could happen in the TOE.

These events contain the following attributes which are important in the context of this evaluation:

Attribute	Description
SPID	ID of the session on which the event occurred.
StartTime	Time at which the event started, if available.
LoginName	Name of the login of the user (either the SQL Server security login or the Microsoft Windows login credentials in the form of DOMAIN\username).
DBUserName	SQL Server database user name of the client.
SessionLoginName	The login name of the user who originated the session. For example, if you connect to SQL Server using Login1 and execute a statement as Login2, SessionLoginName displays Login1, while LoginName displays Login2. This data column displays both SQL Server and Windows logins.
Success	1 = success. 0 = failure. For example, a value of 1 indicates success of a permissions check and a value of 0 indicates failure of that check.
EventSubClass	Type of event subclass., 1=Grant, 2=Revoke, 3=Deny
DatabaseName	Name of the database in which the user statement is running. (if available)
ParentName	Name of the schema the object is within. (if available)
ObjectName	Name of the target object

Table 18: Important attributes of “Audit Object GDR” events

More detailed information about these events can be found in:

Audit Database Scope GDR:

[AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_4deptrbl/html/1641a38a-ef24-46ce-b2f4-bf732858c771.htm]

Audit Schema Object GDR:

[AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_4deptrbl/html/a0187811-dc71-4792-a282-3bfe1ca90c21.htm]

Audit Server Scope GDR Event:

[AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_4deptrbl/html/d3b1e47f-2ba2-49af-b404-1aa231d4e4a0.htm]

Audit Server Object GDR:

[AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_4deptrbl/html/117fedca-c1c4-469a-929a-9ea332c83d25.htm]

Audit Database Object GDR Event:

[AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_4deptrbl/html/2289aab5-e048-4288-bcae-aaf768ca014a.htm]

6.3.6.2 For subjects

Unsuccessful revocation of security attributes of subjects in the context of the evaluated version of the SQL Server 2008 R2 database engine could mean:

- Revoke the group membership of logins or database users (see 6.3.7.3)
- Delete a database user or login (see 6.3.7.1 and 6.3.7.2)

6.3.7 Use of Management functions/Modifications of groups

The following chapters introduce the events which can be used to trace the use of the management functions of the TOE.

6.3.7.1 Add/delete logins

The “Audit Server Principal Management” event class occurs when server principals are created, altered, or dropped. Server principals include all logins and server scoped roles.

This event contains the following attributes which are important in the context of this evaluation:

Attribute	Description
SPID	ID of the session on which the event occurred.
StartTime	Time at which the event started, if available.
LoginName	Name of the login of the user (either the SQL Server security login or the Microsoft Windows login credentials in the form of DOMAIN\username).
DBUserName	SQL Server database user name of the client.
SessionLoginName	The login name of the user who originated the session. For example, if you connect to SQL Server using Login1 and execute a statement as Login2, SessionLoginName displays Login1, while LoginName displays Login2. This data column displays both SQL Server and Windows logins.
Success	1 = success. 0 = failure. For example, a value of 1 indicates success of a permissions check and a value of 0 indicates failure of that check.
EventSubClass	Type of event subclass. 1=Create, 2=Alter, 3=Drop, 4=Dump, 5=Disable, 6=Enable, 11=Load
TextData	Additional information about the principal which is managed in form of a SQL string. This text field also contains information of which type a login is (SQL or Windows) for the case that a login is created
ObjectName	Name of the object being referenced.

Table 19: Important attributes of “Audit Server Principal Management” event

More information about this event can be found in [AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_4deptrbl/html/7894850c-91fe-47c0-a03c-baacbc10d29c.htm]

6.3.7.2 Add/Delete users:

The “Audit Database Principal Management event” class occurs when database principals, such as users, are created, altered, or dropped from a database. Database principals comprise database users and database scoped groups.

This event contains the following attributes which are important in the context of this evaluation:

Attribute	Description
SPID	ID of the session on which the event occurred.
StartTime	Time at which the event started, if available.
LoginName	Name of the login of the user (either the SQL Server security login or the Microsoft Windows login credentials in the form of DOMAIN\username).
DBUserName	SQL Server database user name of the client.
SessionLoginName	The login name of the user who originated the session. For example, if you connect to SQL Server using Login1 and execute a statement as Login2, SessionLoginName displays Login1, while LoginName displays Login2. This data column displays both SQL Server and Windows logins.
Success	1 = success. 0 = failure. For example, a value of 1 indicates success of a permissions check and a value of 0 indicates failure of that check.
EventSubClass	Type of event subclass. 1=Create, 2=Alter, 3=Drop, 4=Dump, 11=Load
TextData	Additional information about the principal which is managed in form of a SQL string.
ObjectName	Name of the object being referenced.

Table 20: Important attributes of “Audit Database Principal Management” event

More information about this event can be found in [AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_4deptrbl/html/594eec78-677c-4500-ae9b-e400abf6f39c.htm]

6.3.7.3 Add and delete group membership for database and server scoped groups

The following events cover the use of this management functionality:

- “Audit Add Login to Server Role” Event Class indicates that a login was added or removed from a fixed server role.
- “Audit Add Member to DB Role” Event Class indicates that a login has been added to or removed from a database role.

The “Audit Add Login to Server Role” event has the following attributes which are important in the context of this evaluation:

Attribute	Description
SPID	ID of the session on which the event occurred.
StartTime	Time at which the event started, if available.
LoginName	Name of the login of the user (either the SQL Server security login or the Microsoft Windows login credentials in the form of DOMAIN\username).
DBUserName	SQL Server database user name of the client.
SessionLoginName	The login name of the user who originated the session. For example, if you connect to SQL Server using Login1 and execute a statement as Login2, SessionLoginName displays Login1, while LoginName displays Login2. This data column displays both SQL Server and Windows logins.
Success	1 = success. 0 = failure. For example, a value of 1 indicates success of a permissions check and a value of 0 indicates failure of that check.
EventSubClass	Type of event subclass. 1=Add, 2=Drop
RoleName	Name of the fixed server role whose membership is being modified.

Table 21: Important attributes of “Audit Add Login to Server Role” event

The “Audit Add Member to DB Role” event has the following attributes which are important in the context of this evaluation:

Attribute	Description
SPID	ID of the session on which the event occurred.
StartTime	Time at which the event started, if available.
LoginName	Name of the login of the user (either the SQL Server security login or the Microsoft Windows login credentials in the form of DOMAIN\username).
DBUserName	SQL Server database user name of the client.
SessionLoginName	The login name of the user who originated the session. For example, if you connect to SQL Server using Login1 and execute a statement as Login2, SessionLoginName displays Login1, while LoginName displays Login2. This data column displays both SQL Server and Windows logins.
Success	1 = success. 0 = failure. For example, a value of 1 indicates success of a permissions check and a value of 0 indicates failure of that check.
EventSubClass	Type of event subclass. 1=Add, 2=Drop, 3=Change group
RoleName	Name of an application role being enabled.

Table 22: Important attributes of “Audit Add Member to DB Role” event

More detailed information about these events can be found in:

“Audit Add Login to Server Role” event:

[AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_4deptrbl/html/7a8ed1c3-a98f-4f93-a6ba-e3901d941db9.htm]

“Audit Add Member to DB Role” event:

[AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_4deptrbl/html/a5ac46b6-765b-4424-b6c7-4d5a1b898d65.htm]

6.3.7.4 Create and destroy database groups

See chapter 6.3.7.2.

6.3.7.5 Start and stop the audit process

See chapter 6.3.2.

6.3.7.6 Include and exclude auditable events.

It is not possible to include or exclude auditable events from or to a trace process while this trace process is running. One has to stop the trace process, apply a filter and start the trace process again. The trace files will contain a Stop and a Start event to indicate that it has been stopped and started again.

See chapter 6.3.4 for more information.

6.3.7.7 Define mode of authentication for every login

The type of a login is defined at creation time. It cannot be changed afterwards. See also chapter 6.3.7.1 as it describes the event which can be captured when a new login is created.

6.3.7.8 Define the action to take in case the audit file is full.

See chapter 6.3.4.

6.3.8 Use of the authentication mechanism

Every use of the authentication mechanism is covered by the use of the following two events:

- “Audit Login” which indicates that a user has successfully logged into SQL Server.
- “Audit Login Failed” which indicates that a user attempted to log in to SQL Server and failed.

The Audit Login event contains the following attributes which are important in the context of this evaluation:

Attribute	Description
SPID	ID of the session on which the event occurred.
StartTime	Time at which the event started, if available.
LoginName	Name of the login of the user (either the SQL Server security login or the Microsoft Windows login credentials in the form of DOMAIN\username).
SessionLoginName	The login name of the user who originated the session. For example, if you connect to SQL Server using Login1 and execute a statement as Login2, SessionLoginName displays Login1, while LoginName displays Login2. This data column displays both SQL Server and Windows logins.
Success	1 = success. 0 = failure. This event will always show success.
TextData	Semicolon-delimited list of all set options.

Table 23: Important attributes of “Audit Login” event

The Audit Login Failed event contains the following attributes which are important in the context of this evaluation:

Attribute	Description
SPID	ID of the session on which the event occurred.
StartTime	Time at which the event started, if available.
LoginName	Name of the login of the user (either the SQL Server security login or the Microsoft Windows login credentials in the form of DOMAIN\username).
SessionLoginName	The login name of the user who originated the session. For example, if you connect to SQL Server using Login1 and execute a statement as Login2, SessionLoginName displays Login1, while LoginName displays Login2. This data column displays both SQL Server and Windows logins.
Success	1 = success. 0 = failure. This event will always show failure.
TextData	Text value dependent on the event class captured in the trace.

Table 24: Important attributes of “Audit Login Failed” event

More detailed information can be found in:

Audit Login:

[AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_4deptrbl/html/ad0bdb48-7f9f-4335-805d-7769d6df89b2.htm]

Audit Login Failed:

[AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_4deptrbl/html/6b83963b-b685-429d-92ba-5173f6f0000d.htm]

6.3.9 Execution of Stored Procedures

Some stored procedures fire relevant events using the User Error Message event class. As every user is in principle able to fire such a User Error Message those messages can in principle not be considered to be safe against spoofing. For this reason the events “sp:starting” and “sp:completed” have been added to the CC trace file definition. This will allow the administrator to see whether a User Error Message has really been fired by a Stored Procedure (in which case the User Error Message will occur between the two corresponding “sp:starting” and “sp:completed” events and the SPID column of sp:starting and sp:completed will show the same ID as for User Error Message).

These events contain the following attributes which are important in the context of this evaluation:

Attribute	Description
SPID	ID of the session on which the event occurred.
StartTime	Time at which the event started, if available.
LoginName	Name of the login of the user (either the SQL Server security login or the Microsoft Windows login credentials in the form of DOMAIN\username).
SessionLoginName	The login name of the user who originated the session. For example, if you connect to SQL Server using Login1 and execute a statement as Login2, SessionLoginName displays Login1, while LoginName displays Login2. This data column displays both SQL Server and Windows logins.
DatabaseName	Name of the database in which the user statement is running. (if available)
TextData	Text of the procedure call

Table 25: Important attributes of sp:starting and sp:completed

For more information about the events please refer to [AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_4deptrbl/html/ef55e579-080d-4650-a7fc-4dd03ed8e391.htm] and [AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_4deptrbl/html/7636a433-5d32-4562-8f5a-694f8e2beeca.htm].

6.3.10 User Error Message

The startup of the TOE is logged using the User Error Message as described in this chapter as early as possible (i.e. directly after the trace has been started). This event is fired by the script that also installs the CC trace process (see chapter 6.3).

This event contains the following attributes which are important in the context of this evaluation:

Attribute	Description
SPID	ID of the session on which the event occurred.
StartTime	Time at which the event started, if available.
LoginName	Name of the login of the user (either the SQL Server security login or the Microsoft Windows login credentials in the form of DOMAIN\username).
SessionLoginName	The login name of the user who originated the session. For example, if you connect to SQL Server using Login1 and execute a statement as Login2, SessionLoginName displays Login1, while LoginName displays Login2. This data column displays both SQL Server and Windows logins.
TextData	Text of the error message or exception

Table 26: Important attributes of “Audit User Error Message” event

The trace script that installs the necessary trace process for the certified version will throw this event after the database engine has been started and the trace process has been started as this is the point of time by which the database engine is operating in its certified configuration.

The TextData of this event will contain the following string: “SQL Server 2008 R2 started (in CC compliant mode), CC trace process started.” Please note that this event is only thrown after a successful startup. As such the column ‘success’ that shows for many events whether it has been a successful operation is not necessary in this context.

For more information about the event please refer to [AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_4deptrbl/html/d7594261-ccd9-487c-9678-11875ba57fb7.htm]

6.4 Deeper audit

It should be noted that the trace functionality of the Microsoft SQL Server 2008 R2 offers many more event classes than the events listed and described in the previous chapters. Further for most of the events additional columns with more detailed information are available.

This allows an administrator to perform an even deeper audit than required in the context of this evaluation.

It would even be possible to log an audit event every time a SQL-Statement has been executed. (Audit SQL:StmtCompleted Event, see [AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_4deptrbl/html/a55f005d-e020-423c-8940-c24ea1b20104.htm]).

However the administrator should consider that a deeper audit will produce bigger trace files and that sufficient disc space for the trace files should be available.

6.5 Filtering of audit and prevention of audit loss

The TOE provides the authorized administrator with the possibility to include or exclude auditable events based on:

- a) user identity and/or group identity,
- b) object identity,
- c) success or failure of auditable security events;

To include or exclude events based on these attributes one has to:

1. Stop the trace process
2. Apply a filter to the trace process
3. Start the trace process again

To apply the filter the Stored Procedure “sp_trace_setfilter” can be used. Please see [AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_6tsql/html/11e7c7ac-a581-4a64-bb15-9272d5c1f7ac.htm] for more details.

6.6 Security Relevant Events

The trace capabilities of the TOE are a powerful mechanism to detect potential security breaches. However the secure operation of the TOE needs the attention of the administrator. He shall review the trace files regularly and pay attention to any suspicious events or events that require a certain action.

As the definition of “suspicious” depends on the concrete installation and environment of the TOE it is not possible to provide a comprehensive definition of what suspicious events are. For example 1000 unsuccessful authentication attempts or failed read attempts per hour may not be suspicious in an installation that serves millions of users while it would be highly suspicious in installations with only a few users.

Classical suspicious events could e.g. be

- An unusual high amount of unsuccessful authentication attempts, which could point to a brute force attack.
- An unusual high amount of events recorded in the trace files could be an indication for an attacker, who is trying to flood the trace files in order to conceal an unauthorized operation.

Other events that can require an action by the administrator include:

- A trace file that is running out of disc space. Depending on the concrete configuration the database engine may shut down or overwrite old trace files if a certain size is reached. In those cases the administrators shall consider to backup the trace files and start over the trace process with a new set of files.
- There are multiple event types in a company that may require the administrator to change settings of the database engine. A classical example is a user owning a

login in the database engine who is leaving the company. In such a case the administrator would usually consider to delete or block the login of the user.

7 Recommendations and requirements for secure administration, configuration and usage

The administrator of the TOE shall follow the following recommendations and requirements to ensure a secure operation of the TOE:

7.1 Recommendations/requirements about Security Audit

- It is recommended to use a separate trace process (also called “CC trace”) to audit all the events which have to be captured according to [ST]. See also chapter 3.3.8 for further guidance to create this trace process. As the CC trace directly starts after the TOE services are available, there can be a small time frame of approx. 2 seconds in which the CC trace does not log incoming connections.
- The CC trace process should always be running. If it is necessary to stop the trace process while the TOE is still running (e.g. to change the configuration of the trace process) it should be considered to create a new trace process which contains all the relevant events as listed in chapter 6.3 and to start this new process before the CC trace process is stopped. In this way it can be ensured that the admin misses no important event.
- For the “CC trace” process it has to be ensured that the option “SHUTDOWN_ON_ERROR” is used, i.e. that the TOE will stop operation in case an error occurs. This option can be combined with the option “TRACE_FILE_ROLLOVER”.
- For the case that the TRACE_FILE_ROLLOVER option is used it is possible that an attacker floods the audit and intentionally causes an event to be overwritten. Thus the administrator has to ensure that sufficient disc space is available for the trace files and appropriate settings are used for the trace processes. Specifically – in cases where the audit of certain event is more important than the availability of the server – it should be considered not to use the TRACE_FILE_ROLLOVER option (i.e. to ensure that the server will shut down if the trace file is full) for all or certain trace processes.

7.2 Recommendations and further information about Access Control

- It should be mentioned that some permissions of the database engine of SQL Server do imply other permissions. A good example of such a permission is the CONTROL SERVER permission that covers all other permissions. The complete hierarchy of permissions within the SQL Server database engine is contained in the file permission_hierarchy.zip ([PERM]) that can be downloaded from [WEB]. This file contains 4 charts that show the permission hierarchy on the 4 levels: server, database, object and column. In each chart it is shown, which permissions on an

object imply which other permissions. For example the extract shown in Figure 19 (from Column.pdf) shows that permissions on a table do automatically imply permissions on the columns of the table. More specific: The CONTROL permission on a table implies permissions for Reference, Select and Update on the columns of the table.

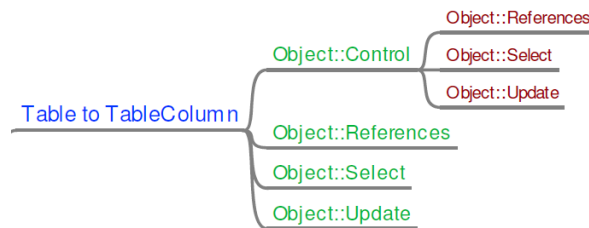


Figure 19: Extract of permission hierarchy

- According to the concept for Access Control in SQL Server 2008 R2 it is possible (if not likely) that two users/administrators have the same permission for one object. This could lead into a situation, where administrators/users cause conflicting operation (e.g. that one administrator grants access to an object while a second administrator denies the same access). These situations can only be avoided by organizational mechanisms and the administrator should be well aware of this fact.
- In its default configuration the database engine of SQL Server 2008 R2 grants the EXECUTE permission on many Stored Procedures to public. This has been done to ensure a maximum level of compatibility to applications. However, some of the Stored Procedures do provide access to sensitive information or open channels for potential attacks. Therefore the administrator shall consider to revoke the EXECUTE permission on all Stored Procedures from public and grant those EXECUTE permissions to specific users or their corresponding groups if necessary.
- The internal access control functionality of the Stored Procedures 'sp_replsendtoqueue' and 'sp_replwritetovarbin' is not compliant to [PP]. Therefore these two procedures must not be accessible by any user within the scope of the certified version of the database engine. After a default installation however the execute permission on these Stored Procedures is granted to public. Therefore the administrator shall revoke the execute permissions from these Stored Procedures from public.
- The Stored Procedure sp_fetchLOBfromcookie that ships together with the database engine has not been considered during the evaluation. As this Stored Procedure is not part of the evaluated version of the database engine the administrator shall revoke the execute permissions on these Stored Procedures from public to ensure that they are not used.
- The description of the sp_dropsvrolemember in [AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_6tsql/html/7be99181-d221-49d0-9cb2-c930d8c044a0.htm] describes that the membership in the sysadmin fixed

server role, or both ALTER ANY LOGIN permission on the server and membership in the role from which the member is being dropped. However to successfully execute this Stored Procedure the pure membership in the role from which a user should be removed is sufficient. The administrator should be aware of the fact that a login who is added to a server role does in this way implicitly inherit the permission to remove all other logins from that role.

- The description of the CREATE LOGIN statement in [AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_6tsql/html/eb737149-7c92-4552-946b-91085d8b1b01.htm] describes that the ALTER ANY LOGIN permission on the server is needed. However – as an exception – the CREATE LOGIN statement can also be executed by a user to create a login for her own Windows account (in this case the user would have access due to the membership in a Windows group).
- The descriptions around the T-SQL commands for creating, altering and dropping database audit specifications in [AGD] lists a set of different permissions that are associated with those commands. However to successfully execute those command the ALTER ANY DATABASE AUDIT permission is the minimum required permission.
- The descriptions around the T-SQL commands for creating, altering and dropping server audit specifications in [AGD] lists a set of different permissions that are associated with those commands. However to successfully execute those command the ALTER ANY SERVER AUDIT permission is the minimum required permission.

7.3 Recommendations/requirements about Identification and Authentication (Secure Passwords)

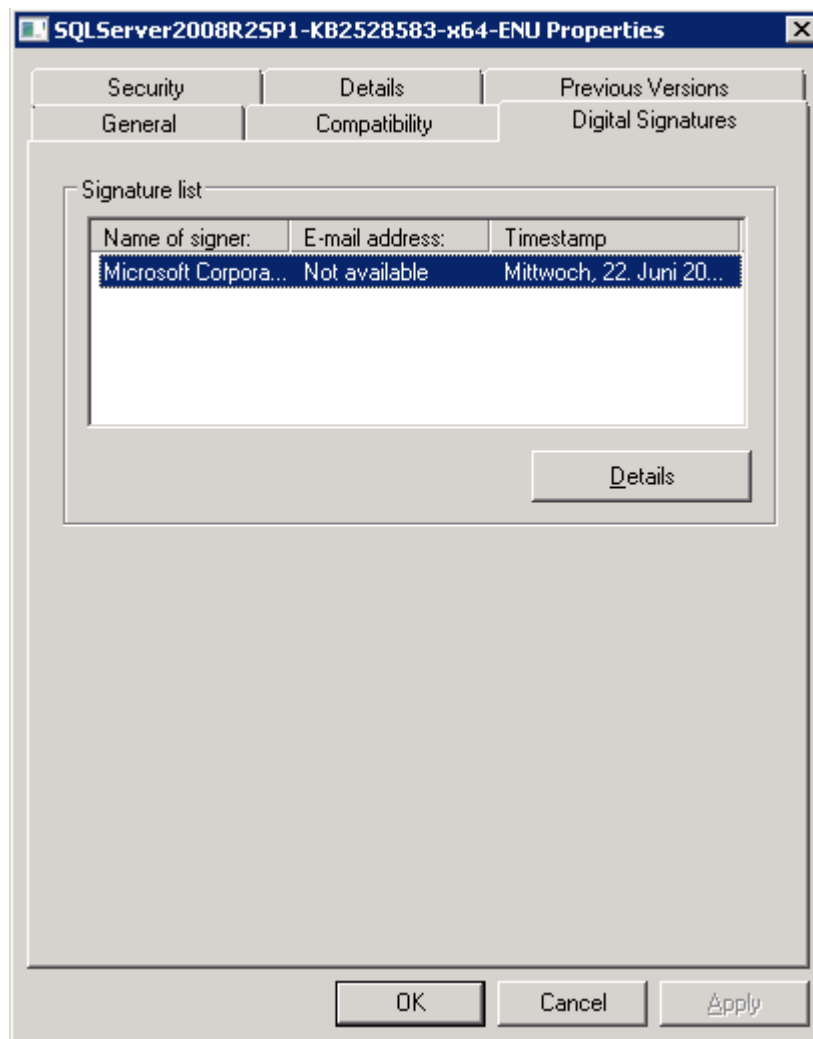
- The administrator(s) shall ensure that passwords for all accounts (service accounts, user accounts and administrative accounts) are of sufficient quality. General guidance, how to create strong passwords can be found under <https://www.microsoft.com/security/pc-security/password-checker.aspx>.
- The concrete settings for the enforcement of minimum password requirements on the underlying Operating System depend on the concrete installation. To allow the secure operation of the TOE the administrator shall ensure that the OS enforces strong password using not less than the following settings:
 - Password must be at least 8 characters in length
 - "password must meet complexity requirements" setting of the OS is enabled. This will ensure that passwords:
 - Do not contain all or part of the user's account name
 - Contain characters from three of the following four categories:
 - English uppercase characters (A through Z)
 - English lowercase characters (a through z)

- Base 10 digits (0 through 9)
- Non-alphabetic characters (for example, !, \$, #, %)
- The SQL Server engine supports the enforcement of password policies for SQL Server logins based on the policies of the underlying Operating System. This option shall be enabled by using the ALTER or CREATE LOGIN command for each login as follows: 'CHECK_POLICY=on'.

7.4 Other Recommendations and requirements

- It is recommended that beside the accounts that are necessary for the administration of the database engine no accounts are created on the machine that the database engine is operating on. Specifically there shall not be any user accounts for users of the database engine that would allow a direct access to the Operating System.
- It should be noted that any changes to logins that occur while a user is connected to the database engine may require the user to log off and log on again before the updated settings take effect. The administrator should therefore consider to terminate a user session (using the KILL command, see also [AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_6tsql/html/071cf260-c794-4b45-adc0-0e64097938c0.htm]) in case of important changes to the login of that user (e.g. the change of group memberships of a user). Further it is possible that sessions are cached after a user disconnected and that a cached session may be reused in case a user logs in again. Changes to login may not be applied to cached sessions under certain circumstances. To avoid this behaviour the administrator shall consider to run the command "DBCC FREESYSTEMCACHE 'ALL'" after important changes to one or more logins. If the server is involved in scenarios of distributed queries the administrator shall further consider to run the "DBCC FREESESSIONCACHE" command in those cases.
- The TOE supports connections via the VIA protocol. However this connection protocol has not been considered during the evaluation. Thus this protocol (which is disabled by default) should not be enabled.
- The Service Broker and Database Mirroring endpoints can be used to circumvent the Security Functionality of the TOE. Therefore the administrator shall not install applications on the TOE that make the TSF or any data controlled by the TSF accessible through these endpoints.
- Per default the connections to the database engine are not encrypted and the encryption features of SQL Server 2008 R2 have not been considered during the evaluation. Thus the administrator has to ensure that all connections to the database engine are appropriately protected, e.g. by using and enforcing an encrypted connection or by using a physically secured connection.

- The use of the column data types text/ntext and image is a deprecated feature (see also [AGD, ms-help://MS.SQLCC.v10/MS.SQLSVR.v10.en/s10de_6tsql/html/b0d8769c-7598-4f97-8162-ace5f182b5bc.htm]) and has not been considered during the evaluation and certification process with respect to the access control functionality. Therefore the administrator shall ensure that user defined objects do not use this data type. The following SQL query can be used to show all columns that use this data type within the current database.
select b.name, a.name from sys.columns a inner join sys.objects b on a.object_id = b.object_id where b.is_ms_shipped=0 and (a.user_type_id=35 or a.user_type_id=99 or a.user_type_id=34)
- Please note that it is possible that after the evaluation and certification process of the TOE as described in this document additional security patches are issued. Therefore the administrator shall regularly visit the Microsoft technet website (<http://www.microsoft.com/technet/security/current.aspx>) to get informed about new security bulletins. For each new security patch the administrator shall carefully consider to install it (depending on the needs of the specific installation). The authenticity of each downloadable package can be verified using the digital signature of the file: a file can be considered authentic if it is digitally signed by Microsoft Corporation (see Figure 20 for an example).
- The Microsoft technet also has a site that explains, how the development group of Microsoft products can be contacted for the case that an administrator finds a security bug (<https://www.microsoft.com/technet/security/bulletin/alertus.aspx>).

**Figure 20: Signature list of SP1**

8 Appendix

8.1 Stored Procedures

The following chapters contain information on Stored Procedures that are contained in SQL Server 2008 R2 but not documented in [AGD].

All these Stored Procedures have been developed for internal use only and are documented for information purposes only. These Stored Procedures are not officially supported by Microsoft and no future compatibility is guaranteed.

8.1.1 sp_MSgetversion

This Stored Procedure can be used to get the current version of Microsoft SQL Server.

Input: no input parameters

Returns: 0 / Error number

Output: row(s) with the Version Number in Character_value

Syntax: exec sp_MSgetversion

8.1.2 xp_dirtree

Returns a complete listing of all subdirectories on the server; for each subdirectory listed its depth in the directory tree is also returned. If a *depth* is specified then only subdirectories up to and including the specified depth will be returned. If *IncludeFiles* is specified (as a 1) then files will also be returned and the result set will include an additional column to indicate if a row is a file or a directory.

Input: @filepath, @depth, @IncludeFiles

Output: subdirectory, depth, file

Note: file is only displayed if @IncludeFiles = 1

Permission If the calling user is 'sa' this Stored Procedure is executed in the context of the SQL Server system account. In all other cases the Stored Procedure will be executed in the context of the calling user (i.e. the Stored Procedure will impersonate the user). This impersonation will fail for the case that a SQL login is used and an empty set will be returned.

Syntax: xp_dirtree <filepath>, <depth>, <IncludeFiles>

Examples: exec xp_dirtree 'c:' - Lists all dirs and sub-dirs on C:

exec xp_dirtree 'c:', 1 - Lists all dirs at the root level of C:

exec xp_dirtree 'c:', 1, 1 - Lists all dirs and files at the root level of C:

8.1.3 xp_fileexist

This Stored Procedure can be used to determine whether a particular file exists on disk or not.

Input: <filename>
Result: 0 / Error number
Permission If the calling user is 'sa' this Stored Procedure is executed in the context of the SQL Server system account. In all other cases the Stored Procedure will be executed in the context of the calling user (i.e. the Stored Procedure will impersonate the user). This impersonation will fail for the case that a SQL login is used and an empty set will be returned.
Syntax: EXECUTE xp_fileexist <filename> [, <file_exists INT> OUTPUT]
Example: For example, to check whether the file boot.ini exists on disk c: or not, run:
EXEC master..xp_fileexist 'c:\boot.ini'

8.1.4 xp_fixeddrives

Returns a row for each fixed drive containing the drive name and the amount of disk space available in MB.

Input: no input parameters
Output: (two columns – drive, MB free)
Permission If the calling user is 'sa' this Stored Procedure is executed in the context of the SQL Server system account. In all other cases the Stored Procedure will be executed in the context of the calling user (i.e. the Stored Procedure will impersonate the user). This impersonation will fail for the case that a SQL login is used and an empty set will be returned.
Syntax: exec @retval=xp_fileexist
Example: To see the list of drives, run:
EXEC master.xp_fixeddrives

8.1.5 xp_getnetname

This extended stored procedure returns the WINS name of the SQL Server that you're connected to.

Input: no input parameters
Output: (optional) one column (Server Net Name)
Else single-row, single-column result set is returned
Syntax: exec @retval=xp_getnetname

8.1.6 xp_MSADEnabled

This Stored Procedure can be used to determine whether the server is on Win NT4 SP5 or later with AD enabled

Input: no input parameters
Result: 0 / Error number

Output: if platform = win32_nt then
 if version > 4 then
 if service pack version > 4 then
 return TRUE;

8.1.7 xp_qv

This Stored Procedure wraps SQLBOOT's QueryProductValue function.

USAGE: xp_qv '<setting>' [, '<instancename>']
 If the optional instance name is not provided, then the default instance ('MSSQLSERVER') is assumed.

RETURNS: A signed int return value from QueryProductValue or VALUE_ERROR (-1), if an error occurred. VALUE_NOT_FOUND (-2) is returned if the input value is not a valid VALUE_* const.

Example: declare @sqlbootvalue int
 exec @sqlbootvalue = xp_qv '2745196162'
 select @sqlbootvalue 'VALUE_REPLICATION'

8.1.8 xp_instance_regread

See xp_regread for details

8.1.9 xp_regread

Functionality: This Stored Procedure is used to read from the registry.

Input: @rootkey, @key, @value_name, [, @value] (can have 5 input parameters)

Comments: Error if <2 input parameters
 5th param – "no_output" then no output is displayed
 No error check if >5 params are given

Permission If the calling user is 'sa' this Stored Procedure is executed in the context of the SQL Server system account. The Stored Procedure ensures that other users are only granted access to a limited set of registry values.

Return: 0/ Error number

Syntax: EXECUTE xp_regread [@rootkey=]'rootkey', [@key=]'key' [,
 @value_name=]'value_name'] [, [@value=]@value OUTPUT]

Example: To read into the variable @test from the value 'TestValue' from the key 'SOFTWARE\Test' from the 'HKEY_LOCAL_MACHINE', run:
 DECLARE @test varchar(20)

```
EXEC master..xp_regread @rootkey='HKEY_LOCAL_MACHINE',
@key='SOFTWARE\Test', @value_name='TestValue', @value=@test
OUTPUT
SELECT @test
```

8.1.10 sp_enable_sql_debug

Functionality: Returns a marshaled COM interface pointer that implements IHostDebugServerInstance, as varbinary(8000). IHostDebugServerInstance is the entry point to the integrated Transact-SQL/CLR debugging interfaces. A debugger calls sp_enable_sql_debug and then unmarshals the returned blob to get IHostDebugServerInstance. All methods of IHostDebugServerInstance and related interface implementations in SQL Server verify the caller is 'sa' and return E_ACCESSDENIED if the check fails.

This Stored Procedure has been developed for debugging purposes only and must not be used in a productive environment.

Input: none

Permission Only 'sa' can call this stored procedure; otherwise permission error 300 will be returned.

Syntax: sp_enable_sql_debug @interface_blob output

Example: declare @v varbinary(8000);
exec master.dbo.sp_enable_sql_debug @v output;
select @v

8.2 References

Reference	Title	Version	Date
[ST]	Security Target, SQL Server 2008 R2 Common Criteria Evaluation	1.04	2011-09-26
[AGD]	SQL Server Books Online	(installed with the TOE)	
[WEB]	https://www.microsoft.com/sqlserver/en/us/common-criteria.aspx (tab "SQL Server 2008 R2 SP1")	-	-
[PERM]	permission_hierarchy.zip, Available via [WEB]		
[PP]	U.S. Government Protection Profile for Database Management Systems in Basic Robustness Environments	1.3	December 24, 2010