Security requirement

Microsoft SQL Server

Deutsche Telekom Group

Version	7.0
Date	Jul 1, 2024
Status	Released

Publication Details

Published by
Deutsche Telekom AG
Vorstandsbereich Technology & Innovation
Chief Security Officer

Reuterstrasse 65, 53315 Bonn
Germany

File name	Document number	Document type
	3.30	Security requirement
Version	**State**	**Status**
7.0	Jul 1, 2024	Released
Contact	**Validity**	**Released by**
Telekom Security psa.telekom.de	Jul 1, 2024 - Jun 30, 2029	Stefan Pütz, Leiter SEC-T-TST

Summary
Microsoft SQL Server

Copyright © 2024 by Deutsche Telekom AG.
All rights reserved.

Table of Contents

1.	Introduction	4
2.	Basic security requirements for Microsoft SQL Server	5
3.	Database system hardening	6
3.1.	Users and roles	6
3.2.	Principle of least privilege	7
3.3.	SQL functions and packages	11
4.	Data communication	13
5.	Misuse detection and prevention	14
6.	MS SQL-specific requirements	20

1. Introduction

This security document has been prepared based on the general security policies of the group.
The security requirement is used as a basis for an approval in the PSA process, among other things. It also serves as an implementation standard for units which do not participate in the PSA process. These requirements shall be taken into account from the very beginning, including during the planning and decision-making processes.
When implementing these security requirements, the precedence of national, international and supranational law shall be observed.

2. Basic security requirements for Microsoft SQL Server

The sections below describe the security requirements which apply especially to MS SQL Server.

Req 1	The edition of the MS SQL Server must be approved for production licensing.

Implementation example: Using the following command, the administrator can determine which edition is installed in the MS SQL Server Management Studio:

```
SELECT SERVERPROPERTY ('edition')
```

ID: 3.30-1/7.0

Req 2	The SQL Server version must be supported by Microsoft for security patching.

Motivation: Only SQL Server versions in mainstream or extended support are supported for security patches.

ID: 3.30-2/7.0

3. Database system hardening

This section contains the requirements for hardening the database system. The measures are comparable with those relating to operating system hardening and reduce the threat of an attack on the database systems.

3.1. Users and roles

Req 3	The standard database administrator account (sa) must be disabled.

Motivation: A large number of automated malware (such as worms) attempts to compromise the default administrator account (sa) using brute force attacks in order to gain unauthorized access to the database. Disabling the account is a simple but effective preventative measure against automated attacks.

ID: 3.30-3/7.0

Req 4	The Windows Built-in accounts or groups must not be SQL Logins.

Motivation: Best practices promote creating an AD level group as a login with sysadmin privileges containing only infrastructure DBA staff accounts as opposed to using the BUILTIN\Administrators group which may contain other non-DBA support staff. It is also recommended to not allow the usage of any of the "Builtin" groups (Everyone,Authenticated Users, Guests, etc). The "Builtin" groups generally contain very broad memberships which would not meet the best practice of ensuring only the necessary users have been granted access to an instance.

Implementation example: The group can be deleted by running the following stored procedure from the "sysadmin" role:

```
EXEC sp_dropsrvrolemember 'BUILTIN\Administrators' , 'sysadmin';
GO
```

ID: 3.30-4/7.0

Req 5	Windows local groups must not be SQL Logins.

Motivation: It is recommended to not allow the usage of any application defined local windows groups (except those created by the Microsoft SQL Server installation for the purpose of providing SQL Services with appropriate permissions such as SQLServer2005MSSQLUser$ComputerName$InstanceName). Allowing local windows groups as SQL Logins provides a loophole whereby anyone with OS level administrator rights (and no SQL Server rights) could add users to the local Windows groups and thereby give themselves or others access to the SQL Server instance.

ID: 3.30-5/7.0

Req 6	Database access for the guest user must not be enabled in user application databases.

Motivation: Only known users may access the databases. Conversely, the GUEST user is intended explicitly for unknown users.

Implementation example: REVOKE CONNECT FROM GUEST

ID: 3.30-6/7.0

Req 7	Installed sample databases must be removed from the SQL Server instance.

Known "sample" databases (pubs, Northwind, or any AdventureWorks database) must not be installed on production systems and must be removed if found.

ID: 3.30-7/7.0

Req 8	If the Trustworthy property is ON, Database owner of a Trustworthy non-system database must not be in the sysadmin role.

By default, the TRUSTWORTHY database property is OFF for user databases.
Reference:
http://blogs.msdn.com/b/sqlsecurity/archive/2007/12/03/the-trustworhy-bit-database-property-in-sql-server-2005.aspx

ID: 3.30-8/7.0

3.2. Principle of least privilege

Req 9	The MS SQL Server must not be set up on a domain controller.

Motivation: This is a manufacturer recommendation by Microsoft.

ID: 3.30-9/7.0

Req 10	The permissions for users and applications must be limited to the extent necessary to fulfill their tasks.

The permissions on a system must be restricted to such an extent that a user can only access data and use functions that he needs in the context of his work. Appropriate permissions must also be assigned for access to files that are part of the operating system or applications or that are generated by the same (e.g. configuration and logging files).

In addition to access to data, applications and their components must also be executed with the lowest possible permissions. Applications should not be run with administrator or system privileges.

Motivation: If a user is granted too far-reaching permissions on a system, he can access data and applications to an extent that is not necessary for the fulfillment of the assigned tasks. This creates an unnecessarily increased risk in the event of abuse, in particular if the user or his user account is compromised by an attacker.
Applications with too far-reaching permissions can be misused by an attacker to gain or expand unauthorized access to data and system areas in need of protection.

For this requirement the following threats are relevant:
- Unauthorized access to the system
- Unauthorized access or tapping of data
- Unauthorized modification of data
- Unauthorized use of services or resources
- Denial of executed activities

ID: 3.01-23/8.0

Req 11	The MSSQL service account must be a Windows domain account.,MSA, or Virtual account.

Motivation: Every additional service with unnecessary rights increases the risk of attack and the level of damage.

ID: 3.30-11/7.0

Req 12	The SQLAgent service account must be a Windows domain account ,MSA, or Virtual account.

Motivation: Every additional service with unnecessary rights increases the risk of attack and the level of damage.

ID: 3.30-12/7.0

Req 13	The SQLFullText service account must be different from the MSSQL service account.

Reference:
http://msdn.microsoft.com/en-us/library/ms143504.aspx#Use_startup_accounts

ID: 3.30-13/7.0

Req 14	The MSSQL service account, must not be a Windows Administrator .

Motivation: Motivation: Restricting the account rights reduces the risk of a system being compromised.

ID: 3.30-14/7.0

Req 15	The SQLAgent service account must not be a Windows Administrator .

Motivation: Restricting the account rights reduces the risk of a system being compromised.

ID: 3.30-15/7.0

Req 16	The SQL Full-Text Service Account must not be a Windows Administrator .

Motivation: Restricting the account rights reduces the risk of a system being compromised.

ID: 3.30-16/7.0

Req 17	The PUBLIC role in the msdb database must not be granted access to SQL Agent proxies.

Motivation: This would allow all users to utilize the proxy which may have high privileges..

ID: 3.30-17/7.0

Req 18	If a password is used as an authentication attribute, it must be changed after 12 months at the latest.

The maximum permitted usage period for passwords is 12 months.
If a password reaches the maximum permitted usage period, it must be changed.

For this purpose, the system must automatically inform the user about the expired usage period the next time he logs on to the system and immediately guide him through a dialog to change the password. Access to the system must no longer be permitted without a successfully completed password change.
For technical user accounts (M2M or Machine-2-Machine), which are used for the authentication and authorization of systems among themselves or by applications on a system, automated solutions must also be implemented to comply with the permitted usage period for passwords.

Alternatively, if such an automatic mapping of the process for changing the password cannot be implemented, an effective organizational measure must be applied instead, wich ensures a binding manual password change at the end of the permissible period of use.

Motivation: Unlike more modern authentication attributes, passwords are easier to attack. Without specific measures for reliable, technically automated detection of compromises, the risk of a password being discovered or broken by an attacker can increase considerably over time.

For this requirement the following threats are relevant:
- Unauthorized access to the system

- Unauthorized access or tapping of data
- Unauthorized modification of data
- Unauthorized use of services or resources
- Attacks motivated and facilitated by information disclosure or visible security weaknesses

ID: 3.01-30/8.0

Req 19	If passwords are used as an authentication attribute, those must be stored using a suitable and approved "Password Hashing" method to protect against offline-attacks like brute force or dictionary attacks.

This requirement relates to the storage of passwords in all types of user databases, as used in this system, in order to authenticate incoming access (local or remote) by users or other systems.

If an attacker obtains the copy of a user database of the system, he is able to bring it into a fully independent environment and utilize automatized dictionary or brute force attacks to determine contained passwords. Specialized tools in combination with high computing power allow for high cracking rates in a relatively short period of time, if protective measures are insufficient. Due to the independency from the source system, such an offline attack happens unnoticed.

The following countermeasure must be implemented, since this ensures best possible protection against offline attacks:

- passwords must be stored using a cryptographic one-way function (known as "Password Hashing") that is recognized as effective for this purpose and is classified as secure at the current state of the art.

Please Note: valid password hashing algorithms are described in Security Requirement Catalog "3.50 Cryptographic Algorithms and Security Protocols".

Explicitly NOT PERMISSIBLE is:

- to store passwords in cleartext
- to store passwords in any format which can be directly backcalculated
- to store passwords using reversible encryption

Please Note:
In this context, "directly backcalculatable formats" refers to those that simply encode the password, without involving a secret key in the transformation process. Since the password will no longer show up as original cleartext after it has been processed, those formats may easily be mistaken to provide confidentiality. Effectively, they do not offer any protection. The enconding is fixed and therefore an attacker can easily make use of it to compute the original cleartext password from the encoded string.
Examples for directly backcalculatable formats are: "base64", "rot13"
"Reversible" are all encryption methods which, using the appropriate key, enable encrypted content to be transformed back into the original content. Accordingly, with reversible encryption there is always the challenge of keeping the key secure and protecting it from unauthorized access. Reversibility is a required fundamental property in many areas of encryption applications, e.g. for transferring confidential messages, but it is counterproductive for storing passwords: a stored password must remain comparable by means of technical methods, but it must no longer be possible to convert it back into plain text in order to protect it as well as possible from unauthorized viewing.
Examples for reversible encryption are: "AES", "CHACHA20", "3DES", "RSA"

Motivation: Without protective measures, an attacker in possession of a user database copy is able to determine masses of contained passwords in short time by merely trying out character string combinations or making use of dictionaries. Passwords stored in cleartext or any backcalculatable format are fully defenseless to an offline attack. Once a password has been ascertained it can be used by an attacker for unauthorized access to the system and the data on it.

For this requirement the following threats are relevant:
- Unauthorized access to the system
- Unauthorized access or tapping of data

- Unauthorized modification of data
- Unauthorized use of services or resources
- Denial of executed activities
- Attacks motivated and facilitated by information disclosure or visible security weaknesses

ID: 3.01-24/8.0

Req 20	If a password is used as an authentication attribute, it must have at least 12 characters and contain three of the following categories: lower-case letters, upper-case letters, digits and special characters.

A system may only accept passwords that comply with the following complexity rules:

- Minimum length of 12 characters.
- Comprising at least three of the following four character categories:
    - lower-case letters
    - upper-case letters
    - digits
    - special characters

The usable maximum length of passwords shall not be limited to less then 25 characters. This will provide more freedom to End Users when composing individual memorizable passwords and helps to prevent undesired behavior in password handling.

When a password is assigned, the system must ensure that the password meets these policies. This must be preferably enforced by technical measures; if such cannot be implemented, organizational measures must be established.
If a central system is used for user authentication [see also Root Security Requirements Document[i] "3.69 IAM (Identity Access Management) - Framework"], it is valid to forward or delegate this task to that central system.

Permissible deviation in the password minimum length
Under security-related criteria, conditions can potentially be identified for a system that enable the minimum password length to be reduced:

- It is generally permissible to reduce the minimum password length for systems that use additional independent authentication attributes within the authentication process in addition to the password (implementation of 2-Factor or Multi-Factor Authentication).
- Any reduction in the minimum password length must be assessed individually by a technical security advisor (e. g. a PSM from Telekom Security) and confirmed as permissible. In the assessment, the surrounding technical, organizational and legal framework parameters must be taken into account, as well as the system-specific protection requirements and the potential amount of damage in the event of security incidents.
- The absolute minimum value of 8 characters length for passwords must not be undercut.

Motivation: Passwords with the above complexity offer contemporary robustness against attacks coupled with acceptable user friendliness. Passwords with this level of complexity have proven their efficiency in practice. Trivial and short passwords are susceptible to brute force and dictionary attacks and are therefore easy for attackers to determine. Once a password has been ascertained it can be used by an attacker for unauthorized access to the system and the data on it.

For this requirement the following threats are relevant:
- Unauthorized access to the system
- Unauthorized access or tapping of data
- Unauthorized modification of data
- Unauthorized use of services or resources
- Denial of executed activities

ID: 3.01-26/8.0

Req 21	If a password is used as an authentication attribute for technical accounts, it must have at least 30 characters and contain three of the following categories: lower-case letters, upper-case letters, digits and special characters.

Technical user accounts are characterized by the fact that they are not used by people. Instead, they are used to authenticate and authorize systems to each other or applications on a system.

A system must only use passwords for technical user accounts that meet the following complexity:

- Minimum length of 30 characters
- Comprising at least three of the following four character categories:
    - lower-case letters
    - upper-case letters
    - digits
    - special characters

Motivation: Due to their use in machine-to-machine (M2M) communication scenarios, technical user accounts are often equipped with privileges that can be of high interest to an attacker to compromise infrastructures. Without mechanisms of extensive compromise detection, the risk of a password being determined or broken by an attacker can increase significantly over time. A significant increase in password length counteracts these risks and can also be implemented particularly easily in M2M scenarios, since handling a very long password is not a particular challenge for a machine (as opposed to a person).

For this requirement the following threats are relevant:

- Unauthorized access to the system
- Unauthorized access or tapping of data
- Unauthorized modification of data
- Unauthorized use of services or resources
- Denial of executed activities

ID: 3.01-27/8.0

3.3. SQL functions and packages

Req 22	Access to all OLE automation stored procedures must be deactivated.

The administrator must deactivate the following OLE automation stored procedures:

- Sp_OACreate,
- Sp_OADestroy,
- SP_OAGetErrorInfo,
- Sp_OAGetProperty,
- SP_OAMethod,
- SP_OASetProperty and
- SP_OAStop.

Motivation: Every unnecessary functionality increases the risk of a successful attack on the system.

Implementation example: sp_configure 'show advanced options', 1;
GO
RECONFIGURE;
GO
sp_configure 'Ole Automation Procedures', 0;

GO RECONFIGURE;
GO

Run the following command to check whether the OLE automation stored procedures are deactivated:

```
EXEC sp_configure ‘Ole Automation Procedures’;
GO
```

ID: 3.30-22/7.0

Req 23	SQL Mail must be deactivated.

Motivation: Every unnecessary functionality increases the risk of a successful attack on the system.

Implementation example: USE [master]
GO
EXECUTE sp_configure 'SQL Mail XPs', 0;
RECONFIGURE;
GO

ID: 3.30-23/7.0

Req 24	If not used, Database Mail must be deactivated.

Implementation example: Database Mail is deactivated be default.

USE [master]
GO
EXECUTE sp_configure 'Database Mail XPs', 0;
RECONFIGURE;
GO

For this requirement the following threats are relevant:
- Unauthorized access to the system

ID: 3.30-24/7.0

Req 25	If SSL is used for access to a database instance, the option “ForceEncryption” must be set to “YES” on the server side.

Motivation: The use of SSL can effectively protect the transfer of confidential data.

ID: 3.30-25/7.0

4. Data communication

This chapter summarizes the requirements concerning the querying of data from other database systems and the exchange of data between database systems. This can be implemented either by means of so-called ad-hoc queries, or by setting a data connection to share data between the systems. The terminology and the technical implementation of such data connections varies greatly among manufacturers (MS-SQL: linked servers/replication; Oracle: database links; DB2: connections). To simplify matters, the term “database links” is used below in a general sense.

Req 26	The “Remote Server” function must be deactivated.

In the more recent SQL Server versions, “Remote Servers” are only supported for the purpose of backward compatibility. More recent applications must use “Linked Servers” instead. “Linked Servers” have better security functions for querying and executing stored procedures on remote instances of MS SQL Servers and other OLE database data sources (e.g., Oracle, Access, Excel, DB2). “Remote Servers” support the concept of remote stored procedures (RPC). The “Remote Server” option must be activated on both the local and the remote server so that a connection can be successfully established. With MS SQL Server version 2005 or later, the “Remote Server” option is deactivated by default for security reasons. The administrator must also activate the MS SQL Server Browser service.

Motivation: The function represents a possible gateway for attackers, as with every functionality which can be accessed remotely.

Implementation example: The stored procedure SP_CONFIGURE can be used to deactivate the “Remote Server”:

```
EXECUTE sp_configure ‘remote access’, 0
RECONFIGURE
GO
```

ID: 3.30-26/7.0

Req 27	If the “Linked Server” function is used to access non-SQL Server providers, the default login mapping for the “Public” role must be deleted.

Motivation: The “Public” role represents a default role for all users. It provides potential attacks with a number of approaches.

Implementation example: By default, all “Linked Servers” and “Remote Servers” can see all logins. To deactivate the default login mapping, the stored procedure “sp_droplinkedsrvlogin” must be executed with NULL (zero) as the “local login” parameter.

```
EXEC sp_droplinkedsrvlogin 'linked-server', NULL;
```

ID: 3.30-27/7.0

Req 28	The named pipes network protocol must be disabled if not used.

Microsoft Best Practices recommends disabling any protocols not required. TCP/IP is preferred over Named Pipes for WANs and slow networks.

ID: 3.30-28/7.0

5. Misuse detection and prevention

Req 29	Accesses to database systems, as well as critical database procedures and database content must be logged.

Secure, traceable database operation requires important operating information to be logged. This includes, for instance, the logging of failed login attempts to uncover possible intrusion attempts.
Logging of security-relevant user actions shall comply with national legislation currently in force.
When implementing measures resulting from this Requirement, the applicable participation rights of the responsible employee representatives/trade unions as well as the works and collective agreements shall be observed.

For this requirement the following threats are relevant:
- Unnoticeable feasible attacks

ID: 3.16-26/7.0

Req 30	The SQL Server default trace must be enabled.

Motivation: Default tracing provides information about configuration and DDL changes should be logged for , traceable database operation.

ID: 3.30-30/7.0

Req 31	The number of Error log must be increased to at least 12.

By default, the SQL Server only creates up to 7 error logs, a new one is created every restart of the server.

Motivation: Secure, traceable database operation requires important operating information to be logged. Error logs therefore should not be overwritten after 7 restarts of the server.

ID: 3.30-31/7.0

Req 32	The 'xp_cmdshell' Server Configuration Option must be set to 0.

Motivation: Enabling this option can provide the ability for non-sysadmins to execute OS commands.

Implementation example: USE [master]
GO
    EXECUTE sp_configure 'xp_cmdshell', 0;
RECONFIGURE;
GO

ID: 3.30-32/7.0

Req 33	The 'Remote Admin Connections' Server Configuration Option must be set to 0 on non-clustered instances.

Motivation: This option enables access to the dedicated admin connection remotely. However, this option must be enabled on clustered instances.

Implementation example: Do not execute this on a clustered instance.

USE [master]
GO
    EXECUTE sp_configure 'Remote Admin Connections', 0;

```
RECONFIGURE;
GO
```

ID: 3.30-33/7.0

Req 34	The 'CLR Assembly Permission Set' must be set to SAFE_ACCESS for all user-defined CLR Assemblies.

Implementation example: To find user created assemblies, execute:

```
SELECT name AS Assembly_Name, permission_set_desc
FROM sys.assemblies
WHERE is_user_defined = 1
and permission_set_desc <> 'SAFE_ACCESS';
```

For compliance, no rows should be returned.

How to remediate:
For each Assembly_Name returned in the query above, execute:

```
USE [master]
GO
ALTER ASSEMBLY Assembly_Name WITH PERMISSION_SET = SAFE;
GO
```

This should first be tested within a test environment prior to production to ensure the assembly still functions as designed with SAFE permission setting.

ID: 3.30-34/7.0

Req 35	Security-related events must be logged with an exact timestamp and a unique system reference.

Security-relevant are any events that may have an impact on the information security (confidentiality, integrity, availability) of a system or indicate its misuse. Systems must log the occurrence of security-related events.
In order to enable the assignment and evaluation of these events, they must be logged together with a unique system name (e.g. host name, IP or MAC address) and the exact time of occurrence ("time stamp"). At a minimum, the timestamp of a logged event must include the following information:

- date of the event (Year, Month, Day)
- time of the event (Hours, Minutes, Seconds)
- Timezone, those information belongs to

The security relevance of specific events depends on the type and function of the system as well as the data processed in it. Therefore, it is not possible to specify a uniform list of events that are relevant to security. Relevant events must be identified according to the system. The following list contains typical security-related events and is accordingly to be understood as a basic guide for the identification of security-relevant events for a particular system.

Security-related Event	Information to log to the event
Failed login attempts	• affected user account • number of failed attempts • source of access (IP address, client-ID/client-name, local console, ...)

System access via user accounts (including system accounts or technical user accounts)	• affected user account • Access start/end time (login/logout) • source of access (IP address, client-ID/client-name, local console, ...)
Modifications of user accounts (including system accounts or technical user accounts)	• user account used to make the modification • user account to which the modification was made • type of modification made (create, activate, deactivate, delete, password reset, ...)
Changes to role assignements/permissions of user accounts (including system accounts or technical user accounts)	• user account used to make the modification • user account to which the modification was made • name of the affected role, type of the affected permission • type of modification made (role/permission added, role/permission removed, ...)
Use/initiation of critical system functions	• user account used to make the use • name/type of system function used
Access to sensitive data processed in the system	• user account used to access the data • Name of the affected data field (the contents of the data field should not be logged to prevent data leakage via log files) • type of access (read, modify, delete, ...)
Abnormal increase in the utilization of system resources, such as memory, CPU, network, or disk access, over an extended period of time	• threshold of the affected system resource • duration of exceeding the threshold • threshold exceedance Depending on the individual system, sensible threshold values must be defined.
Other events that may provide evidence of attempted or successful attacks against the system ***)**	• individually depending on the type of event

***)** *This requires identifying the most important forms of attack that the system can typically expect and ensuring that events relevant to their detection are actually logged by the system. As a rule, this can only be guaranteed by the developer/manufacturer of the system.*

When logging, the applicable legal and operational regulations must be observed. The latter also include agreements that have been made with the company's social partners. Following these regulations logging of events is only allowed for a defined use case. Logging of events for doing a work control of employees is not allowed.

In addition - as for any data that is processed by a system - an appropriate protection requirement must also be taken into account and implemented for logging data; this applies to storage, transmission and access. In particular, if the logging data contains real data, the same protection requirements must be taken into account that is also used for the regular processing of this real data within the source system.

Motivation: Logging security-relevant events is a basic requirement for detecting ongoing attacks as well as attacks that have already occurred. This is the only way in which suitable measures can be taken to maintain or restore system security. Logging data could be used as evidence to take legal steps against attackers.

For this requirement the following threats are relevant:
- Denial of executed activities
- Unnoticeable feasible attacks

ID: 3.01-33/8.0

Req 36	Applicable retention and deletion periods must be observed for security-relevant logging data that is recorded locally.

From an IT security perspective, local storage of security-relevant logging data on a system is not mandatory. Since the local storage can be damaged in the event of system malfunctions or manipulated by a successful attacker, it can only be used to a limited extent for security-related or forensic analyses. Accordingly, it is relevant for IT security that logging data is forwarded to a separate log server.

Local storage can nevertheless take place; for example, if local storage is initially indispensable when generating the logging data due to technical processes or if there are justified operational interests in also keeping logging data available locally.

The following basic rules must be taken into account when storing logging data locally:

- Security-related logging data must be retained for a period of 90 days.
  (*This requirement only applies if no additional forwarding to a separate log server is implemented on the system and the logging data is therefore only recorded locally.*)
- After 90 days, stored logging data must be deleted immediately.

Deviances

Different retention periods and deletion periods may exist due to legal or regulatory requirements (especially in connection with personal data) or may be defined by contractual agreements. In these cases, the applicable periods must be agreed individually with a Project Security Manager (PSM) / Data Privacy Advisor (DPA) or are specified by them.

Motivation: Logging data is an immensely important IT security tool for preventing, detecting and clearing up system faults, security and data privacy incidents. On the other hand, the recording of logging data, like any other data processing, is also subject to legal and regulatory requirements. Accordingly, guidelines must be adhered to that reconcile the two.

Implementation example: Taking into account the current legal situation and applicable data privacy regulations, the following deletion periods for locally stored security-relevant logging data are implemented on an exemplary telecommunications system:

- Standard System Logs: Deletion after 90 days at the latest
- Logging of public IP addresses: Deletion (or anonymization) after 7 days at the latest
- Logging of the assignment of dynamic public IP addresses by the telecommunication solution: Deletion after 7 days at the latest
- Logging of non-billing-relevant call detail records: Deletion after 7 days at the latest
- Logging of the content of e-mail and SMS: Deletion after 24 hours at the latest
- Logging of the domain queries handled by the DNS server of the telecommunications solution: Deletion after 24 hours at the latest

For this requirement the following threats are relevant:

- Unauthorized access or tapping of data
- Denial of executed activities
- Unnoticeable feasible attacks

ID: 3.01-34/8.0

Req 37	Security-relevant logging data must be forwarded to a separate log server immediately after it has been generated.

Logging data must be forwarded to a separate log server immediately after it has been generated. Standardized proto-

cols such as Syslog, SNMPv3 should be preferred.

Motivation: If logging data is only stored locally, it can be manipulated by an attacker who succeeds in compromising the system in order to conceal his attack and any manipulation he has performed on the system. This is the reason why the forwarding must be done immediately after the event occurred.

For this requirement the following threats are relevant:
- Unauthorized modification of data
- Disruption of availability
- Denial of executed activities
- Unnoticeable feasible attacks

ID: 3.01-35/8.0

Req 38	For security-relevant logging data that is forwarded to the separate log server, compliance with the applicable retention and deletion periods must be ensured.

The following basic rules must be taken into account:
- security-related logging data must be retained for a period of 90 days on the separate log server.
- after 90 days, stored logging data must be deleted immediately on the separate log server.

Deviances
Different retention periods and deletion periods may exist due to legal or regulatory requirements (especially in connection with personal data) or may be defined by contractual agreements. In these cases, the applicable periods must be agreed individually with a Project Security Manager (PSM) / Data Privacy Advisor (DSB) or are specified by them.

Log server under the responsibility of a third party
If the selected separate log server is not within the same operational responsibility as the source system of the loggin data, it must be ensured that the responsible operator of the log server is aware of the valid parameters for the logging data to be received and that they are adhered to in accordance with the regulations mentioned here.

Motivation: Logging data is an immensely important IT security tool for preventing, detecting and clearing up system faults, security and data privacy incidents. On the other hand, the recording of logging data, like any other data processing, is also subject to legal and regulatory requirements. Accordingly, guidelines must be adhered to that reconcile the two.

Implementation example: Taking into account the current legal situation and applicable data privacy regulations, the following deletion periods for forwarded security-relevant logging data from an exemplary telecommunications system are implemented on the separate log server:
- Standard System Logs: Deletion after 90 days at the latest
- Logging of public IP addresses: Deletion (or anonymization) after 7 days at the latest
- Logging of the assignment of dynamic public IP addresses by the telecommunication solution: Deletion after 7 days at the latest
- Logging of non-billing-relevant call detail records: Deletion after 7 days at the latest
- Logging of the content of e-mail and SMS: Deletion after 24 hours at the latest
- Logging of the domain queries handled by the DNS server of the telecommunications solution: Deletion after 24 hours at the latest

For this requirement the following threats are relevant:
- Unauthorized access or tapping of data
- Denial of executed activities
- Unnoticeable feasible attacks

ID: 3.01-36/8.0

Req 39	Security-relevant logging data must be provided to a security incident detection and response solution in near-real-time.

An acceptable detection and response solution for the purposes of this requirement must meet the following criteria:

- There is a continuous analysis and correlation of the logging data provided by connected IT/NT systems.
- The identification of threatening activities and ongoing attacks takes place on the basis of a continuously updated detection pattern database
- Integrated alerting processes trigger the initiation of countermeasures in the event of a find.

Such a solution is not the same as a log server in terms of its function. Accordingly, the provision to a detection and response solution must always be in addition to an existing forwarding to a log server. An exception exists if the log server used provides the logging data to a detection and response solution immediately after receipt. In this case, it is sufficient for the system to forward the logging data only to the log server.
The logging data must be provided to the detection and response solution within 5 minutes ("near-real-time") from the time it is generated in the system.

The specific technical interfaces and protocols to be supported for deployment are determined by the detection and response solution available in the system's operating environment as well as the technical basis of the system itself.

Motivation: IT/NT systems are constantly exposed to security threats and attacks. By means of a central analysis and the correlation of recorded system events of entire IT landscapes in almost real time, current threats and ongoing attacks can be detected even beyond system boundaries and appropriate countermeasures can be initiated promptly. The prompt delivery of logging data to the detection and response solution is essential for its effectiveness.

Implementation example: **Example 1:** A system based on a classic server collects security-relevant logging data in the operating system's local syslog or Windows Event Log. Logging data is continuously forwarded to a Security Information and Event Management (SIEM) for further analysis, intrusion detection, and alarm handling.

Example 2: An Endpoint Detection and Response (EDR) agent is running on a system. The EDR agent continuously forwards the security-related logging data it collects on the system to the central backend of the EDR solution, where further analysis, intrusion detection, and alarm handling takes place

Example 3: Systems designed on the basis of cloud-native functions generate logging data directly from these functions using the cloud's own logging and monitoring infrastructure. In such a system, the generation of security-relevant logging data is specifically activated in the corresponding configuration options of the cloud-native functions used. The resulting logging data is continuously transferred to a SIEM (Security Information and Event Management) for further analysis, intrusion detection and alarm handling. Some cloud providers already provide some attack pattern detection in the cloud's own monitoring infrastructure and generate specific cloud alerts from it. These are also transferred to the SIEM.

Example 4: A system is based on a SaaS ("Software as a Service") application from a cloud provider. Typically, in the SaaS model, the cloud provider is responsible for all operational aspects of the application, including security management. The cloud customer itself has no influence on the underlying technical infrastructure or the operation of the application. Based on contractual agreements, recognized certificates, and documented self-disclosures from the cloud provider, it is formally verified that the cloud provider has implemented a security incident detection and response solution. Some cloud providers also generate dedicated logging data about security-relevant activities within the separate customer tenants of their SaaS application (e.g. changes to configuration settings, permission assignments or user accounts within the tenants) and make them available to cloud customers. These so-called audit logs are continuously transferred to a SIEM (Security Information and Event Management) for further analysis, attack detection and alarm handling.

For this requirement the following threats are relevant:

- Unnoticeable feasible attacks

ID: 3.01-37/8.0

6. MS SQL-specific requirements

In addition to the security requirements derived from the main document, the manufacturer-specific requirements listed below should be taken into account.

Req 40	Windows authentication mode must be used.

Motivation: Compared with Windows authentication, mixed-mode authentication provides an inadequate standard of security. Windows authentication uses the Kerberos protocol. In addition, the validation of password complexity, account blocking and the password workflow can be enforced using Group policies.

ID: 3.30-40/7.0

Req 41	The 'Cross DB Ownership Chaining' Server Configuration Option must be set to 0.

Motivation: The activated configuration option applies to all databases running on the instance. The area open to attack will be increased unnecessarily for all databases as a result.

ID: 3.30-41/7.0

Req 42	The 'DB_CHAINING' Database Property Setting must be set to OFF.

Motivation: Restricting the rights reduces the area open to attack and thus increases security.

Implementation example: ALTER DATABASE dbname SET DB_CHAINING OFF;
GO

ID: 3.30-42/7.0