

**CIS Microsoft SQL Server 2019 Benchmark**

v1.2.0

-

04

-

30

-

2021

# Terms of Use

Please see the below link for our current terms of use:

*https://www.cisecurity.org/cis-securesuite/cis-securesuite-membership-terms-of-use/*

Table of Contents

[Terms of Use 1](#_Toc150388)

[Overview 5](#_Toc150389)

[Intended Audience 5](#_Toc150390)

[Consensus Guidance 5](#_Toc150391)

[Typographical Conventions 6](#_Toc150392)

[Assessment Status 6](#_Toc150393)

[Profile Definitions 7](#_Toc150394)

[Acknowledgements 8](#_Toc150395)

[Recommendations 9](#_Toc150396)

[1 Installation, Updates and Patches 9](#_Toc150397)

[1.1 Ensure Latest SQL Server Cumulative and Security Updates are Installed](#_Toc150398)

[(Manual) 9](#_Toc150399)

[1.2 Ensure Single-Function Member Servers are Used (Manual) 11](#_Toc150400)

[2 Surface Area Reduction 13](#_Toc150401)

[2.1 Ensure 'Ad Hoc Distributed Queries' Server Configuration Option is set to '0' (Automated) 13](#_Toc150402)

[2.2 Ensure 'CLR Enabled' Server Configuration Option is set to '0' (Automated) 15](#_Toc150403)

[2.3 Ensure 'Cross DB Ownership Chaining' Server Configuration Option is set to '0' (Automated) 18](#_Toc150404)

[2.4 Ensure 'Database Mail XPs' Server Configuration Option is set to '0'](#_Toc150405)

[(Automated) 20](#_Toc150406)

[2.5 Ensure 'Ole Automation Procedures' Server Configuration Option is set to '0' (Automated) 22](#_Toc150407)

[2.6 Ensure 'Remote Access' Server Configuration Option is set to '0' (Automated) 24](#_Toc150408)

[2.7 Ensure 'Remote Admin Connections' Server Configuration Option is set to '0' (Automated) 26](#_Toc150409)

[2.8 Ensure 'Scan For Startup Procs' Server Configuration Option is set to '0'](#_Toc150410)

[(Automated) 28](#_Toc150411)

[2.9 Ensure 'Trustworthy' Database Property is set to 'Off' (Automated) 30](#_Toc150412)

[2.10 Ensure Unnecessary SQL Server Protocols are set to 'Disabled' (Manual) 32](#_Toc150413)

[2.11 Ensure SQL Server is configured to use non-standard ports (Automated) 34](#_Toc150414)

[2.12 Ensure 'Hide Instance' option is set to 'Yes' for Production SQL Server](#_Toc150415)

[instances (Automated) 36](#_Toc150416)

[2.13 Ensure the 'sa' Login Account is set to 'Disabled' (Automated) 39](#_Toc150417)

[2.14 Ensure the 'sa' Login Account has been renamed (Automated) 41](#_Toc150418)

[2.15 Ensure 'AUTO\_CLOSE' is set to 'OFF' on contained databases (Automated) 43](#_Toc150419)

[2.16 Ensure no login exists with the name 'sa' (Automated) 45](#_Toc150420)

[2.17 Ensure 'clr strict security' Server Configuration Option is set to '1'](#_Toc150421)

[(Automated) 47](#_Toc150422)

[3 Authentication and Authorization 49](#_Toc150423)

[3.1 Ensure 'Server Authentication' Property is set to 'Windows Authentication](#_Toc150424)

[Mode' (Automated) 49](#_Toc150425)

[3.2 Ensure CONNECT permissions on the 'guest' user is Revoked within all SQL Server databases excluding the master, msdb and tempdb (Automated) 51](#_Toc150426)

[3.3 Ensure 'Orphaned Users' are Dropped From SQL Server Databases](#_Toc150427)

[(Automated) 53](#_Toc150428)

[3.4 Ensure SQL Authentication is not used in contained databases (Automated) 55](#_Toc150429)

[3.5 Ensure the SQL Server’s MSSQL Service Account is Not an Administrator](#_Toc150430)

[(Manual) 57](#_Toc150431)

[3.6 Ensure the SQL Server’s SQLAgent Service Account is Not an Administrator](#_Toc150432)

[(Manual) 59](#_Toc150433)

[3.7 Ensure the SQL Server’s Full-Text Service Account is Not an Administrator](#_Toc150434)

[(Manual) 61](#_Toc150435)

[3.8 Ensure only the default permissions specified by Microsoft are granted to the public server role (Automated) 63](#_Toc150436)

[3.9 Ensure Windows BUILTIN groups are not SQL Logins (Automated) 65](#_Toc150437)

[3.10 Ensure Windows local groups are not SQL Logins (Automated) 67](#_Toc150438)

[3.11 Ensure the public role in the msdb database is not granted access to SQL Agent proxies (Automated) 69](#_Toc150439)

[4 Password Policies 71](#_Toc150440)

[4.1 Ensure 'MUST\_CHANGE' Option is set to 'ON' for All SQL Authenticated Logins (Manual) 71](#_Toc150441)

[4.2 Ensure 'CHECK\_EXPIRATION' Option is set to 'ON' for All SQL Authenticated](#_Toc150442)

[Logins Within the Sysadmin Role (Automated) 73](#_Toc150443)

[4.3 Ensure 'CHECK\_POLICY' Option is set to 'ON' for All SQL Authenticated Logins (Automated) 75](#_Toc150444)

[5 Auditing and Logging 77](#_Toc150445)

[5.1 Ensure 'Maximum number of error log files' is set to greater than or equal to '12' (Automated) 77](#_Toc150446)

[5.2 Ensure 'Default Trace Enabled' Server Configuration Option is set to '1'](#_Toc150447)

[(Automated) 80](#_Toc150448)

[5.3 Ensure 'Login Auditing' is set to 'failed logins' (Automated) 82](#_Toc150449)

[5.4 Ensure 'SQL Server Audit' is set to capture both 'failed' and 'successful logins' (Automated) 85](#_Toc150450)

[6 Application Development 88](#_Toc150451)

[6.1 Ensure Database and Application User Input is Sanitized (Manual) 88](#_Toc150452)

[6.2 Ensure 'CLR Assembly Permission Set' is set to 'SAFE\_ACCESS' for All CLR](#_Toc150453)

[Assemblies (Automated) 90](#_Toc150454)

[7 Encryption 92](#_Toc150455)

[7.1 Ensure 'Symmetric Key encryption algorithm' is set to 'AES\_128' or higher in non-system databases (Automated) 92](#_Toc150456)

[7.2 Ensure Asymmetric Key Size is set to 'greater than or equal to 2048' in non-system databases (Automated) 94](#_Toc150457)

[8 Appendix: Additional Considerations 96](#_Toc150458)

[8.1 Ensure 'SQL Server Browser Service' is configured correctly (Manual) 96](#_Toc150459)

[Appendix: Recommendation Summary Table 98](#_Toc150460)

[Appendix: Change History 101](#_Toc150461)

# Overview

This document provides prescriptive guidance for establishing a secure configuration posture for Microsoft SQL Server 2019. This guide was tested against Microsoft SQL Server

2019. To obtain the latest version of this guide, please visit http://benchmarks.cisecurity.org. If you have questions, comments, or have identified ways to improve this guide, please write us at feedback@cisecurity.org.

## Intended Audience

This benchmark is intended for system and application administrators, security specialists, auditors, help desk, and platform deployment personnel who plan to develop, deploy, assess, or secure solutions that incorporate Microsoft SQL Server 2019 on a Microsoft Windows platform.

## Consensus Guidance

This benchmark was created using a consensus review process comprised of subject matter experts. Consensus participants provide perspective from a diverse set of backgrounds including consulting, software development, audit and compliance, security research, operations, government, and legal.

Each CIS benchmark undergoes two phases of consensus review. The first phase occurs during initial benchmark development. During this phase, subject matter experts convene to discuss, create, and test working drafts of the benchmark. This discussion occurs until consensus has been reached on benchmark recommendations. The second phase begins after the benchmark has been published. During this phase, all feedback provided by the Internet community is reviewed by the consensus team for incorporation in the benchmark. If you are interested in participating in the consensus process, please visit https://workbench.cisecurity.org/.

## Typographical Conventions

The following typographical conventions are used throughout this guide:

|  |  |  |
| --- | --- | --- |
| Convention | | Meaning |
|  | Stylized Monospace font | Used for blocks of code, command, and script examples. Text should be interpreted exactly as presented. |
|  |
| Monospace font | | Used for inline code, commands, or examples. Text should be interpreted exactly as presented. |
| *<italic font in brackets>* | | Italic texts set in angle brackets denote a variable requiring substitution for a real value. |
| *Italic font* | | Used to denote the title of a book, article, or other publication. |
| **Note** | | Additional information or caveats |

## Assessment Status

An assessment status is included for every recommendation. The assessment status indicates whether the given recommendation can be automated or requires manual steps to implement. Both statuses are equally important and are determined and supported as defined below:

**Automated**

Represents recommendations for which assessment of a technical control can be fully automated and validated to a pass/fail state. Recommendations will include the necessary information to implement automation.

**Manual**

Represents recommendations for which assessment of a technical control cannot be fully automated and requires all or some manual steps to validate that the configured state is set as expected. The expected state can vary depending on the environment.

## Profile Definitions

The following configuration profiles are defined by this Benchmark:

* **Level 1 - Database Engine**

Items in this profile apply to Microsoft SQL Server 2019 and intend to:

* + be practical and prudent; o provide a clear security benefit; and o not inhibit the utility of the technology beyond acceptable means.
* **Level 1 - AWS RDS**

Items in this profile are applicable to Microsoft SQL Server 2019 on AWS RDS and intend to:

* + be practical and prudent; o provide a clear security benefit; and o not inhibit the utility of the technology beyond acceptable means.
* **Level 2 - Database Engine**

This profile extends the "Level 1 - Database Engine" profile. Items in this profile apply to Microsoft SQL Server 2019 and exhibit one or more of the following characteristics:

* + are intended for environments or use cases where security is paramount; o acts as defense in depth measure; and o may impact the utility or performance of the technology.
* **Level 2 - AWS RDS**

This profile extends the "Level 1 - AWS RDS" profile. Items in this profile apply to Microsoft SQL Server 2019 on AWS RDS and exhibit one or more of the following characteristics:

* + are intended for environments or use cases where security is paramount; o acts as defense in depth measure; and o may impact the utility or performance of the technology.

## Acknowledgements

This benchmark exemplifies the great things a community of users, vendors, and subject matter experts can accomplish through consensus collaboration. The CIS community thanks the entire consensus team with special recognition to the following individuals who contributed greatly to the creation of this guide:

**Contributor**

Michal Horan

Dean Lackey

Matthew Woods

Rob Kraft

Emad Al-Mousa

**Editor**

Brian Kelley MCSE, CISA, Security+, Microsoft MVP - SQL Server

Tim Harrison, Center for Internet Security Sean McCown

# Recommendations

## 1 Installation, Updates and Patches

This section contains recommendations related to installing and patching SQL Server.

#### 1.1 Ensure Latest SQL Server Cumulative and Security Updates are

#### Installed (Manual)

**Profile Applicability:**

* Level 1 - Database Engine
* Level 1 - AWS RDS

**Description:**

SQL Server patches contain program updates that fix security and product functionality issues found in the software. These patches can be installed with a security update, which is a single patch, or a cumulative update which is a group of patches. The SQL Server version and patch levels should be the most recent compatible with the organizations' operational needs.

**Rationale:**

Using the most recent SQL Server software, along with all applicable patches can help limit the possibilities for vulnerabilities in the software. The installation version and/or patches applied during setup should be established according to the needs of the organization.

**Audit:**

To determine your SQL Server patch level, run the following code snippet.

SELECT SERVERPROPERTY('ProductLevel') as SP\_installed, SERVERPROPERTY('ProductVersion') as Version;

**Remediation:**

Identify the current version and patch level of your SQL Server instances and ensure they contain the latest security fixes. Make sure to test these fixes in your test environments before updating production instances.

The most recent SQL Server patches can be found here: https://docs.microsoft.com/en-us/sql/database-engine/install-windows/latest-updatesfor-microsoft-sql-server?view=sql-server-ver15

**Default Value:**

Cumulative and security updates are not installed by default.

**References:**

1. https://docs.microsoft.com/en-us/sql/database-engine/install-windows/latestupdates-for-microsoft-sql-server?view=sql-server-ver15
2. https://support.microsoft.com/en-us/help/4041553/sql-server-service-packs-arediscontinued-starting-from-sql-server

**CIS Controls:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Controls Version** | **Control** | **IG 1** | **IG 2** | **IG 3** |
| v7 | 2.2 Ensure Software is Supported by Vendor  Ensure that only software applications or operating systems currently supported by the software's vendor are added to the organization's authorized software inventory. Unsupported software should be tagged as unsupported in the inventory system. | ● | ● | ● |
| v6 | 4 Continuous Vulnerability Assessment and Remediation  Continuous Vulnerability Assessment and Remediation |  |  |  |

#### 1.2 Ensure Single-Function Member Servers are Used (Manual)

**Profile Applicability:**

* Level 1 - Database Engine
* Level 1 - AWS RDS

**Description:**

It is recommended that SQL Server software be installed on a dedicated server. This architectural consideration affords security flexibility in that the database server can be placed on a separate subnet allowing access only from particular hosts and over particular protocols. Degrees of availability are easier to achieve as well - over time, an enterprise can move from a single database server to a failover to a cluster using load balancing or to some combination thereof.

**Rationale:**

It is easier to manage (i.e. reduce) the attack surface of the server hosting SQL Server software if the only surfaces to consider are the underlying operating system, SQL Server itself, and any security/operational tooling that may additionally be installed. As noted in the description, availability can be more easily addressed if the database is on a dedicated server.

**Impact:**

It is difficult to see any reasonably adverse impact to making this architectural change, once the costs of making the change have been paid. Custom applications may need to be modified to accommodate database connections over the wire rather than on the host (i.e. using TCP/IP instead of Named Pipes). Additional hardware and operating system licenses may be required to make these architectural changes.

**Audit:**

Ensure that no other roles are enabled for the underlying operating system and that no excess tooling is installed, per enterprise policy.

**Remediation:**

Uninstall excess tooling and/or remove unnecessary roles from the underlying operating system.

**CIS Controls:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Controls Version** | **Control** | **IG 1** | **IG 2** | **IG 3** |
| v7 | 2.10 Physically or Logically Segregate High Risk Applications  Physically or logically segregated systems should be used to isolate and run software that is required for business operations but incur higher risk for the organization. |  |  | ● |
| v6 | 9.5 Operate Critical Services On Dedicated Hosts (i.e. DNS, Mail, Web, Database)  Operate critical services on separate physical or logical host machines, such as DNS, file, mail, web, and database servers. |  |  |  |

## 2 Surface Area Reduction

SQL Server offers various configuration options, some of them can be controlled by the sp\_configure stored procedure. This section contains the listing of the corresponding recommendations.

#### 2.1 Ensure 'Ad Hoc Distributed Queries' Server Configuration Option is set to '0' (Automated)

**Profile Applicability:**

* Level 1 - Database Engine
* Level 1 - AWS RDS

**Description:**

Enabling Ad Hoc Distributed Queries allows users to query data and execute statements on external data sources. This functionality should be disabled.

**Rationale:**

This feature can be used to remotely access and exploit vulnerabilities on remote SQL Server instances and to run unsafe Visual Basic for Application functions.

**Audit:**

Run the following T-SQL command:

SELECT name, CAST(value as int) as value\_configured, CAST(value\_in\_use as int) as value\_in\_use FROM sys.configurations

WHERE name = 'Ad Hoc Distributed Queries';

Both value columns must show 0.

**Remediation:**

Run the following T-SQL command:

EXECUTE sp\_configure 'show advanced options', 1;

RECONFIGURE;

EXECUTE sp\_configure 'Ad Hoc Distributed Queries', 0;

RECONFIGURE;

GO

EXECUTE sp\_configure 'show advanced options', 0; RECONFIGURE;

**Default Value:**

0 (disabled)

**References:**

1. https://docs.microsoft.com/en-us/sql/database-engine/configure-windows/adhoc-distributed-queries-server-configuration-option

**CIS Controls:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Controls Version** | **Control** | **IG 1** | **IG 2** | **IG 3** |
| v7 | 9.2 Ensure Only Approved Ports, Protocols and Services Are Running  Ensure that only network ports, protocols, and services listening on a system with validated business needs, are running on each system. |  | ● | ● |
| v6 | 9.1 Limit Open Ports, Protocols, and Services  Ensure that only ports, protocols, and services with validated business needs are running on each system. |  |  |  |

#### 2.2 Ensure 'CLR Enabled' Server Configuration Option is set to '0' (Automated)

**Profile Applicability:**

* Level 1 - Database Engine
* Level 1 - AWS RDS

**Description:**

The clr enabled option specifies whether user assemblies can be run by SQL Server.

**Rationale:**

Enabling use of CLR assemblies widens the attack surface of SQL Server and puts it at risk from both inadvertent and malicious assemblies.

**Impact:**

If CLR assemblies are in use, applications may need to be rearchitected to eliminate their usage before disabling this setting. Alternatively, some organizations may allow this setting to be enabled 1 for assemblies created with the SAFE permission set, but disallow assemblies created with the riskier UNSAFE and EXTERNAL\_ACCESS permission sets. To find user-created assemblies, run the following query in all databases, replacing *<database\_name>* with each database name:

USE [<database\_name>]

GO

SELECT name AS Assembly\_Name, permission\_set\_desc

FROM sys.assemblies

WHERE is\_user\_defined = 1;

GO

**Audit:**

Run the following T-SQL command:

SELECT name,

CAST(value as int) as value\_configured,

CAST(value\_in\_use as int) as value\_in\_use

FROM sys.configurations

WHERE name = 'clr strict security';

If both values are 1, this recommendation is Not Applicable. Otherwise, run the following TSQL command:

SELECT name,

CAST(value as int) as value\_configured,

CAST(value\_in\_use as int) as value\_in\_use

FROM sys.configurations

WHERE name = 'clr enabled';

Both value columns must show 0 to be compliant.

**Remediation:**

Run the following T-SQL command:

EXECUTE sp\_configure 'clr enabled', 0; RECONFIGURE;

**Default Value:**

By default, this option is disabled (0).

**References:**

1. https://docs.microsoft.com/en-us/sql/t-sql/statements/create-assembly-transactsql

**Additional Information:**

If clr strict security is set to 1 this recommendation is not applicable. By default, clr strict security is enabled and treats SAFE and EXTERNAL\_ACCESS assemblies as if they were marked UNSAFE. Though not recommended, the clr strict security option can be disabled for backward compatibility. This recommendation has been retained for environments configured for backwards compatibility.

**CIS Controls:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Controls Version** | **Control** | **IG 1** | **IG 2** | **IG 3** |
| v7 | 18.11 Use Standard Hardening Configuration Templates for Databases  For applications that rely on a database, use standard hardening configuration |  | ● | ● |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Controls Version** | **Control** | **IG 1** | **IG 2** | **IG 3** |
|  | templates. All systems that are part of critical business processes should also be tested. |  |  |  |
| v6 | 18.9 Sanitize Deployed Software Of Development Artifacts  For in-house developed applications, ensure that development artifacts (sample data and scripts; unused libraries, components, debug code; or tools) are not included in the deployed software, or accessible in the production environment. |  |  |  |

#### 2.3 Ensure 'Cross DB Ownership Chaining' Server Configuration Option is set to '0' (Automated)

**Profile Applicability:**

* Level 1 - Database Engine
* Level 1 - AWS RDS

**Description:**

The cross db ownership chaining option controls cross-database ownership chaining across all databases at the instance (or server) level.

**Rationale:**

When enabled, this option allows a member of the db\_owner role in a database to gain access to objects owned by a login in any other database, causing an unnecessary information disclosure. When required, cross-database ownership chaining should only be enabled for the specific databases requiring it instead of at the instance level for all databases by using the ALTER DATABASE*<database\_name>*SET DB\_CHAINING ON command. This database option may not be changed on the master, model, or tempdb system databases.

**Audit:**

Run the following T-SQL command:

SELECT name,

CAST(value as int) as value\_configured,

CAST(value\_in\_use as int) as value\_in\_use

FROM sys.configurations

WHERE name = 'cross db ownership chaining';

Both value columns must show 0 to be compliant.

**Remediation:**

Run the following T-SQL command:

|  |
| --- |
| EXECUTE sp\_configure 'cross db ownership chaining', 0;  RECONFIGURE; GO |

**Default Value:**

By default, this option is disabled (0).

**References:**

1. https://docs.microsoft.com/en-us/sql/database-engine/configure-windows/crossdb-ownership-chaining-server-configuration-option

**CIS Controls:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Controls Version** | **Control** | **IG 1** | **IG 2** | **IG 3** |
| v7 | 14.6 Protect Information through Access Control Lists  Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities. | ● | ● | ● |
| v6 | 14.4 Protect Information With Access Control Lists  All information stored on systems shall be protected with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities. |  |  |  |

#### 2.4 Ensure 'Database Mail XPs' Server Configuration Option is set to '0'

#### (Automated)

**Profile Applicability:**

* Level 1 - Database Engine
* Level 1 - AWS RDS

**Description:**

The Database Mail XPs option controls the ability to generate and transmit email messages from SQL Server.

**Rationale:**

Disabling the Database Mail XPs option reduces the SQL Server surface, eliminates a DOS attack vector and channel to exfiltrate data from the database server to a remote host.

**Audit:**

Run the following T-SQL command:

SELECT name,

CAST(value as int) as value\_configured,

CAST(value\_in\_use as int) as value\_in\_use

FROM sys.configurations

WHERE name = 'Database Mail XPs';

Both value columns must show 0 to be compliant.

**Remediation:**

Run the following T-SQL command:

EXECUTE sp\_configure 'show advanced options', 1;

RECONFIGURE;

EXECUTE sp\_configure 'Database Mail XPs', 0;

RECONFIGURE;

GO

EXECUTE sp\_configure 'show advanced options', 0; RECONFIGURE;

**Default Value:**

By default, this option is disabled (0).

**References:**

1. https://docs.microsoft.com/en-us/sql/relational-databases/databasemail/database-mail

**CIS Controls:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Controls Version** | **Control** | **IG 1** | **IG 2** | **IG 3** |
| v7 | 9.2 Ensure Only Approved Ports, Protocols and Services  Are Running  Ensure that only network ports, protocols, and services listening on a system with validated business needs, are running on each system. |  | ● | ● |
| v6 | 18 Application Software Security  Application Software Security |  |  |  |

#### 2.5 Ensure 'Ole Automation Procedures' Server Configuration Option is set to '0' (Automated)

**Profile Applicability:**

* Level 1 - Database Engine
* Level 1 - AWS RDS

**Description:**

The Ole Automation Procedures option controls whether OLE Automation objects can be instantiated within Transact-SQL batches. These are extended stored procedures that allow SQL Server users to execute functions external to SQL Server.

**Rationale:**

Enabling this option will increase the attack surface of SQL Server and allow users to execute functions in the security context of SQL Server.

**Audit:**

Run the following T-SQL command:

SELECT name,

CAST(value as int) as value\_configured,

CAST(value\_in\_use as int) as value\_in\_use

FROM sys.configurations

WHERE name = 'Ole Automation Procedures';

Both value columns must show 0 to be compliant.

**Remediation:**

Run the following T-SQL command:

EXECUTE sp\_configure 'show advanced options', 1;

RECONFIGURE;

EXECUTE sp\_configure 'Ole Automation Procedures', 0;

RECONFIGURE;

GO

EXECUTE sp\_configure 'show advanced options', 0; RECONFIGURE;

**Default Value:**

By default, this option is disabled (0).

**References:**

1. https://docs.microsoft.com/en-us/sql/database-engine/configure-windows/oleautomation-procedures-server-configuration-option

**CIS Controls:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Controls Version** | **Control** | **IG 1** | **IG 2** | **IG 3** |
| v7 | 5.1 Establish Secure Configurations  Maintain documented, standard security configuration standards for all authorized operating systems and software. | ● | ● | ● |
| v6 | 18 Application Software Security  Application Software Security |  |  |  |

#### 2.6 Ensure 'Remote Access' Server Configuration Option is set to '0' (Automated)

**Profile Applicability:**

* Level 1 - Database Engine
* Level 1 - AWS RDS

**Description:**

The remote access option controls the execution of local stored procedures on remote servers or remote stored procedures on local server.

**Rationale:**

Functionality can be abused to launch a Denial-of-Service (DoS) attack on remote servers by off-loading query processing to a target.

**Impact:**

Per Microsoft: This feature will be removed in the next version of Microsoft SQL Server. Do not use this feature in new development work, and modify applications that currently use this feature as soon as possible. Use sp\_addlinkedserver instead.

**Audit:**

Run the following T-SQL command:

SELECT name,

CAST(value as int) as value\_configured,

CAST(value\_in\_use as int) as value\_in\_use

FROM sys.configurations WHERE name = 'remote access';

Both value columns must show 0.

**Remediation:**

Run the following T-SQL command:

EXECUTE sp\_configure 'show advanced options', 1;

RECONFIGURE;

EXECUTE sp\_configure 'remote access', 0;

RECONFIGURE;

GO

EXECUTE sp\_configure 'show advanced options', 0; RECONFIGURE;

Restart the Database Engine.

**Default Value:**

By default, this option is enabled (1).

**References:**

1. https://docs.microsoft.com/en-us/sql/database-engine/configurewindows/configure-the-remote-access-server-configuration-option

**CIS Controls:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Controls Version** | **Control** | **IG 1** | **IG 2** | **IG 3** |
| v7 | 9.2 Ensure Only Approved Ports, Protocols and Services Are Running  Ensure that only network ports, protocols, and services listening on a system with validated business needs, are running on each system. |  | ● | ● |
| v6 | 9.1 Limit Open Ports, Protocols, and Services  Ensure that only ports, protocols, and services with validated business needs are running on each system. |  |  |  |

#### 2.7 Ensure 'Remote Admin Connections' Server Configuration Option is set to '0' (Automated)

**Profile Applicability:**

* Level 1 - Database Engine
* Level 1 - AWS RDS

**Description:**

The remote admin connections option controls whether a client application on a remote computer can use the Dedicated Administrator Connection (DAC).

**Rationale:**

The Dedicated Administrator Connection (DAC) lets an administrator access a running server to execute diagnostic functions or Transact-SQL statements, or to troubleshoot problems on the server, even when the server is locked or running in an abnormal state and not responding to a SQL Server Database Engine connection. In a cluster scenario, the administrator may not actually be logged on to the same node that is currently hosting the SQL Server instance and thus is considered "remote". Therefore, this setting should usually be enabled (1) for SQL Server failover clusters; otherwise, it should be disabled (0) which is the default.

**Audit:**

Run the following T-SQL command:

USE master;

GO

SELECT name,

CAST(value as int) as value\_configured,

CAST(value\_in\_use as int) as value\_in\_use

FROM sys.configurations

WHERE name = 'remote admin connections'

AND SERVERPROPERTY('IsClustered') = 0;

If no data is returned, the instance is a cluster and this recommendation is not applicable. If data is returned, then both the value columns must show 0 to be compliant.

**Remediation:**

Run the following T-SQL command on non-clustered installations:

|  |
| --- |
| EXECUTE sp\_configure 'remote admin connections', 0;  RECONFIGURE; GO |

**Default Value:**

By default, this option is disabled (0), only local connections may use the DAC.

**References:**

1. https://docs.microsoft.com/en-us/sql/database-engine/configurewindows/remote-admin-connections-server-configuration-option

**Additional Information:**

If it's a clustered installation, this option must be enabled as a clustered SQL Server cannot bind to localhost and DAC will be unavailable otherwise. Enable it for clustered installations. Disable it for standalone installations where not required.

**CIS Controls:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Controls Version** | **Control** | **IG 1** | **IG 2** | **IG 3** |
| v7 | 9.2 Ensure Only Approved Ports, Protocols and Services Are Running  Ensure that only network ports, protocols, and services listening on a system with validated business needs, are running on each system. |  | ● | ● |
| v6 | 9.1 Limit Open Ports, Protocols, and Services  Ensure that only ports, protocols, and services with validated business needs are running on each system. |  |  |  |

#### 2.8 Ensure 'Scan For Startup Procs' Server Configuration Option is set to

#### '0' (Automated)

**Profile Applicability:**

* Level 1 - Database Engine
* Level 1 - AWS RDS

**Description:**

The scan for startup procs option, if enabled, causes SQL Server to scan for and automatically run all stored procedures that are set to execute upon service startup.

**Rationale:**

Enforcing this control reduces the threat of an entity leveraging these facilities for malicious purposes.

**Impact:**

Setting Scan for Startup Procedures to 0 will prevent certain audit traces and other commonly used monitoring stored procedures from re-starting on start up. Additionally, replication requires this setting to be enabled (1) and will automatically change this setting if needed.

**Audit:**

Run the following T-SQL command:

SELECT name,

CAST(value as int) as value\_configured,

CAST(value\_in\_use as int) as value\_in\_use

FROM sys.configurations

WHERE name = 'scan for startup procs';

Both value columns must show 0.

**Remediation:**

Run the following T-SQL command:

EXECUTE sp\_configure 'show advanced options', 1;

RECONFIGURE;

EXECUTE sp\_configure 'scan for startup procs', 0; RECONFIGURE;

GO

EXECUTE sp\_configure 'show advanced options', 0; RECONFIGURE;

Restart the Database Engine.

**Default Value:**

By default, this option is disabled (0).

**References:**

1. https://docs.microsoft.com/en-us/sql/database-engine/configurewindows/configure-the-scan-for-startup-procs-server-configuration-option

**CIS Controls:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Controls Version** | **Control** | **IG 1** | **IG 2** | **IG 3** |
| v7 | 5.1 Establish Secure Configurations  Maintain documented, standard security configuration standards for all authorized operating systems and software. | ● | ● | ● |
| v6 | 18 Application Software Security  Application Software Security |  |  |  |

#### 2.9 Ensure 'Trustworthy' Database Property is set to 'Off' (Automated)

**Profile Applicability:**

• Level 1 - Database Engine

**Description:**

The TRUSTWORTHY database option allows database objects to access objects in other databases under certain circumstances.

**Rationale:**

Provides protection from malicious CLR assemblies or extended procedures.

**Audit:**

Run the following T-SQL query to list any databases with a Trustworthy database property value of ON:

SELECT name

FROM sys.databases

WHERE is\_trustworthy\_on = 1 AND name != 'msdb';

No rows should be returned.

**Remediation:**

Execute the following T-SQL statement against the databases (replace *<database\_name>* below) returned by the Audit Procedure:

ALTER DATABASE [<database\_name>] SET TRUSTWORTHY OFF;

**Default Value:**

By default, this database property is OFF (is\_trustworthy\_on = 0), except for the msdb database in which it is required to be ON.

**References:**

1. https://docs.microsoft.com/en-us/sql/relational-databases/security/trustworthydatabase-property
2. https://support.microsoft.com/it-it/help/2183687/guidelines-for-using-thetrustworthy-database-setting-in-sql-server

**CIS Controls:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Controls Version** | **Control** | **IG 1** | **IG 2** | **IG 3** |
| v7 | 14.6 Protect Information through Access Control Lists  Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities. | ● | ● | ● |
| v6 | 14.4 Protect Information With Access Control Lists  All information stored on systems shall be protected with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities. |  |  |  |

#### 2.10 Ensure Unnecessary SQL Server Protocols are set to 'Disabled' (Manual)

**Profile Applicability:**

• Level 1 - Database Engine

**Description:**

SQL Server supports Shared Memory, Named Pipes, and TCP/IP protocols. However, SQL Server should be configured to use the bare minimum required based on the organization's needs.

**Rationale:**

Using fewer protocols minimizes the attack surface of SQL Server and, in some cases, can protect it from remote attacks.

**Impact:**

The Database Engine (MSSQL and SQLAgent) services must be stopped and restarted for the change to take effect.

**Audit:**

Open **SQL Server Configuration Manager**; go to the **SQL Server Network Configuration**.

Ensure that only required protocols are enabled.

**Remediation:**

Open **SQL Server Configuration Manager**; go to the **SQL Server Network Configuration**.

Ensure that only required protocols are enabled. Disable protocols not necessary.

**Default Value:**

By default, TCP/IP and Shared Memory protocols are enabled on all commercial editions.

**References:**

1. https://docs.microsoft.com/en-us/sql/database-engine/configurewindows/enable-or-disable-a-server-network-protocol

**CIS Controls:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Controls Version** | **Control** | **IG 1** | **IG 2** | **IG 3** |
| v7 | 9.2 Ensure Only Approved Ports, Protocols and Services Are Running  Ensure that only network ports, protocols, and services listening on a system with validated business needs, are running on each system. |  | ● | ● |
| v6 | 9.1 Limit Open Ports, Protocols, and Services  Ensure that only ports, protocols, and services with validated business needs are running on each system. |  |  |  |

### 2.11 Ensure SQL Server is configured to use non-standard ports (Automated)

**Profile Applicability:**

* Level 1 - Database Engine
* Level 1 - AWS RDS

**Description:**

If installed, a default SQL Server instance will be assigned a default port of TCP:1433 for TCP/IP communication. Administrators can also manually configure named instances to use TCP:1433 for communication. TCP:1433 is a widely known SQL Server port and this port assignment should be changed. In a multi-instance scenario, each instance must be assigned its own dedicated TCP/IP port.

**Rationale:**

Using a non-default port helps protect the database from attacks directed to the default port.

**Impact:**

Changing the default port will force the DAC (Dedicated Administrator Connection) to listen on a random port. Also, it might make benign applications, such as application firewalls, require special configuration. In general, you should set a static port for consistent usage by applications, including firewalls, instead of using dynamic ports which will be chosen randomly at each SQL Server start up.

**Audit:**

Run the one of following T-SQL script:

SELECT TOP(1) local\_tcp\_port FROM sys.dm\_exec\_connections

WHERE local\_tcp\_port IS NOT NULL;

Or

SELECT local\_tcp\_port

FROM sys.dm\_exec\_connections WHERE session\_id = @@SPID

If a value of 1433 is returned this is a fail.

**Remediation:**

1. In **SQL Server Configuration Manager**, in the console pane, expand **SQL Server**

**Network Configuration**, expand Protocols for *<InstanceName>*, and then doubleclick the TCP/IP protocol

1. In the **TCP/IP Properties** dialog box, on the **IP Addresses** tab, several IP addresses appear in the format IP1, IP2, up to IPAll. One of these is for the IP address of the loopback adapter, 127.0.0.1. Additional IP addresses appear for each IP Address on the computer.
2. Under IPAll, change the **TCP Port** field from 1433 to a non-standard port or leave the **TCP Port** field empty and set the **TCP Dynamic Ports** value to 0 to enable dynamic port assignment and then click **OK**.
3. In the console pane, click **SQL Server Services**.
4. In the details pane, right-click **SQL Server (*<InstanceName>*)** and then click **Restart**, to stop and restart SQL Server.

**Default Value:**

By default, default SQL Server instances listen on to TCP/IP traffic on TCP port 1433 and named instances use dynamic ports.

**References:**

1. https://docs.microsoft.com/en-us/sql/database-engine/configurewindows/configure-a-server-to-listen-on-a-specific-tcp-port

**Additional Information:**

In the case of AWS RDS, this is only configurable during the build process.

**CIS Controls:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Controls Version** | **Control** | **IG 1** | **IG 2** | **IG 3** |
| v7 | 9.2 Ensure Only Approved Ports, Protocols and Services  Are Running  Ensure that only network ports, protocols, and services listening on a system with validated business needs, are running on each system. |  | ● | ● |
| v6 | 9 Limitation and Control of Network Ports, Protocols, and Services  Limitation and Control of Network Ports, Protocols, and Services |  |  |  |

### 2.12 Ensure 'Hide Instance' option is set to 'Yes' for Production SQL

### Server instances (Automated)

**Profile Applicability:**

• Level 1 - Database Engine

**Description:**

Non-clustered SQL Server instances within production environments should be designated as hidden to prevent advertisement by the SQL Server Browser service.

**Rationale:**

Designating production SQL Server instances as hidden leads to a more secure installation because they cannot be enumerated. However, clustered instances may break if this option is selected.

**Impact:**

This method only prevents the instance from being listed on the network. If the instance is hidden (not exposed by SQL Browser), then connections will need to specify the server and port in order to connect. It does not prevent users from connecting to server if they know the instance name and port.

If you hide a clustered named instance, the cluster service may not be able to connect to the SQL Server. Please refer to the Microsoft documentation reference.

**Audit:**

Perform either the GUI or T-SQL method shown:

***1.1.1.1 GUI Method***

1. In **SQL Server Configuration Manager**, expand **SQL Server Network Configuration**, right-click **Protocols for *<InstanceName>***, and then select **Properties**.
2. On the **Flags** tab, in the **Hide Instance** box, if Yes is selected, it is compliant.

***1.1.1.2 T-SQL Method***

Execute the following T-SQL.

DECLARE @getValue INT;

EXEC master.sys.xp\_instance\_regread

@rootkey = N'HKEY\_LOCAL\_MACHINE',

@key = N'SOFTWARE\Microsoft\Microsoft SQL

Server\MSSQLServer\SuperSocketNetLib',

@value\_name = N'HideInstance',

@value = @getValue OUTPUT; SELECT @getValue;

A value of 1 should be returned to be compliant.

**Remediation:**

Perform either the GUI or T-SQL method shown:

***1.1.1.3 GUI Method***

1. In **SQL Server Configuration Manager**, expand **SQL Server Network Configuration**, right-click **Protocols for *<InstanceName>***, and then select **Properties**.
2. On the **Flags** tab, in the **Hide Instance** box, select Yes, and then click **OK** to close the dialog box. The change takes effect immediately for new connections.

***1.1.1.4 T-SQL Method***

Execute the following T-SQL to remediate:

EXEC master.sys.xp\_instance\_regwrite

@rootkey = N'HKEY\_LOCAL\_MACHINE',

@key = N'SOFTWARE\Microsoft\Microsoft SQL

Server\MSSQLServer\SuperSocketNetLib',

@value\_name = N'HideInstance',

@type = N'REG\_DWORD', @value = 1;

**Default Value:**

By default, SQL Server instances are not hidden.

**References:**

1. https://docs.microsoft.com/en-us/sql/database-engine/configure-windows/hidean-instance-of-sql-server-database-engine

**CIS Controls:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Controls Version** | **Control** | **IG 1** | **IG 2** | **IG 3** |
| v7 | 9.2 Ensure Only Approved Ports, Protocols and Services  Are Running  Ensure that only network ports, protocols, and services listening on a system with validated business needs, are running on each system. |  | ● | ● |
| v6 | 9 Limitation and Control of Network Ports, Protocols, and Services  Limitation and Control of Network Ports, Protocols, and Services |  |  |  |

### 2.13 Ensure the 'sa' Login Account is set to 'Disabled' (Automated)

**Profile Applicability:**

* Level 1 - Database Engine
* Level 1 - AWS RDS

**Description:**

The sa account is a widely known and often widely used SQL Server account with sysadmin privileges. This is the original login created during installation and always has the principal\_id=1 and sid=0x01.

**Rationale:**

Enforcing this control reduces the probability of an attacker executing brute force attacks against a well-known principal.

**Impact:**

It is not a good security practice to code applications or scripts to use the sa account. However, if this has been done, disabling the sa account will prevent scripts and applications from authenticating to the database server and executing required tasks or functions.

**Audit:**

Use the following syntax to determine if the sa account is disabled. Checking for sid=0x01 ensures that the original sa account is being checked in case it has been renamed per best practices.

SELECT name, is\_disabled

FROM sys.server\_principals

WHERE sid = 0x01

AND is\_disabled = 0;

No rows should be returned to be compliant.

An is\_disabled value of 0 indicates the login is currently enabled and therefore needs remediation.

**Remediation:**

Execute the following T-SQL query:

USE [master]

GO

DECLARE @tsql nvarchar(max)

SET @tsql = 'ALTER LOGIN ' + SUSER\_NAME(0x01) + ' DISABLE'

EXEC (@tsql) GO

**Default Value:**

By default, the sa login account is disabled at install time when Windows Authentication Mode is selected. If mixed mode (SQL Server and Windows Authentication) is selected at install, the default for the sa login is enabled.

**References:**

1. https://docs.microsoft.com/en-us/sql/relational-databases/system-catalogviews/sys-server-principals-transact-sql
2. https://docs.microsoft.com/en-us/sql/t-sql/statements/alter-login-transact-sql
3. https://docs.microsoft.com/en-us/sql/relational-databases/security/choose-anauthentication-mode

**Additional Information:**

In the case of AWS RDS the default name for this account is rdsa instead of sa.

**CIS Controls:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Controls Version** | **Control** | **IG 1** | **IG 2** | **IG 3** |
| v7 | 16.8 Disable Any Unassociated Accounts  Disable any account that cannot be associated with a business process or business owner. | ● | ● | ● |
| v6 | 5.1 Minimize And Sparingly Use Administrative Privileges  Minimize administrative privileges and only use administrative accounts when they are required. Implement focused auditing on the use of administrative privileged functions and monitor for anomalous behavior. |  |  |  |

### 2.14 Ensure the 'sa' Login Account has been renamed (Automated)

**Profile Applicability:**

* Level 1 - Database Engine
* Level 1 - AWS RDS

**Description:**

The sa account is a widely known and often widely used SQL Server login with sysadmin privileges. The sa login is the original login created during installation and always has principal\_id=1 and sid=0x01.

**Rationale:**

It is more difficult to launch password-guessing and brute-force attacks against the sa login if the name is not known.

**Impact:**

It is not a good security practice to code applications or scripts to use the sa login. However, if this has been done, renaming the sa login will prevent scripts and applications from authenticating to the database server and executing required tasks or functions.

**Audit:**

Use the following syntax to determine if the sa login (principal) is renamed.

SELECT name

FROM sys.server\_principals WHERE sid = 0x01;

A name of sa indicates the account has not been renamed and therefore needs remediation. **Remediation:**

Replace the *<different\_user>* value within the below syntax and execute to rename the sa login.

ALTER LOGIN sa WITH NAME = <different\_user>;

**Default Value:**

By default, the sa login name is 'sa'.

**References:**

1. https://docs.microsoft.com/en-us/sql/relational-databases/security/choose-anauthentication-mode

**Additional Information:**

In the case of AWS RDS the default name for this account is rdsa instead of sa.

**CIS Controls:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Controls Version** | **Control** | **IG 1** | **IG 2** | **IG 3** |
| v7 | 5.1 Establish Secure Configurations  Maintain documented, standard security configuration standards for all authorized operating systems and software. | ● | ● | ● |
| v6 | 5 Controlled Use of Administration Privileges  Controlled Use of Administration Privileges |  |  |  |

### 2.15 Ensure 'AUTO\_CLOSE' is set to 'OFF' on contained databases (Automated)

**Profile Applicability:**

* Level 1 - Database Engine
* Level 1 - AWS RDS

**Description:**

AUTO\_CLOSE determines if a given database is closed or not after a connection terminates. If enabled, subsequent connections to the given database will require the database to be reopened and relevant procedure caches to be rebuilt.

**Rationale:**

Because authentication of users for contained databases occurs within the database not at the server\instance level, the database must be opened every time to authenticate a user. The frequent opening/closing of the database consumes additional server resources and may contribute to a denial of service.

**Audit:**

Perform the following to find contained databases that are not configured as prescribed:

SELECT name, containment, containment\_desc, is\_auto\_close\_on

FROM sys.databases

WHERE containment <> 0 and is\_auto\_close\_on = 1;

No rows should be returned.

**Remediation:**

Execute the following T-SQL, replacing *<database\_name>* with each database name found by the Audit Procedure:

ALTER DATABASE <database\_name> SET AUTO\_CLOSE OFF;

**Default Value:**

By default, the database property AUTO\_CLOSE is OFF which is equivalent to is\_auto\_close\_on = 0.

**References:**

1. https://docs.microsoft.com/en-us/sql/relational-databases/databases/securitybest-practices-with-contained-databases

**CIS Controls:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Controls Version** | **Control** | **IG 1** | **IG 2** | **IG 3** |
| v7 | 5.1 Establish Secure Configurations  Maintain documented, standard security configuration standards for all authorized operating systems and software. | ● | ● | ● |
| v6 | 18 Application Software Security  Application Software Security |  |  |  |

### 2.16 Ensure no login exists with the name 'sa' (Automated)

**Profile Applicability:**

* Level 1 - Database Engine
* Level 1 - AWS RDS

**Description:**

The sa login (e.g. principal) is a widely known and often widely used SQL Server account. Therefore, there should not be a login called sa even when the original sa login (principal\_id = 1) has been renamed.

**Rationale:**

Enforcing this control reduces the probability of an attacker executing brute force attacks against a well-known principal name.

**Impact:**

It is not a good security practice to code applications or scripts to use the sa account. Given that it is a best practice to rename and disable the sa account, some 3rd party applications check for the existence of a login named sa and if it doesn't exist, creates one. Removing the sa login will prevent these scripts and applications from authenticating to the database server and executing required tasks or functions.

**Audit:**

Use the following syntax to determine if there is an account named sa.

SELECT principal\_id, name

FROM sys.server\_principals WHERE name = 'sa';

No rows should be returned.

**Remediation:**

Execute the appropriate ALTER or DROP statement below based on the principal\_id returned for the login named sa. Replace the *<different\_name>* value within the below syntax and execute to rename the sa login.

USE [master]

GO

|  |
| --- |
| -- If principal\_id = 1 or the login owns database objects, rename the sa login  ALTER LOGIN [sa] WITH NAME = <different\_name>;  GO  -- If the login owns no database objects, then drop it  -- Do NOT drop the login if it is principal\_id = 1 DROP LOGIN sa |

**Default Value:**

The login with principal\_id = 1 is named sa by default.

**CIS Controls:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Controls Version** | **Control** | **IG 1** | **IG 2** | **IG 3** |
| v7 | 5.1 Establish Secure Configurations  Maintain documented, standard security configuration standards for all authorized operating systems and software. | ● | ● | ● |
| v6 | 5.1 Minimize And Sparingly Use Administrative Privileges  Minimize administrative privileges and only use administrative accounts when they are required. Implement focused auditing on the use of administrative privileged functions and monitor for anomalous behavior. |  |  |  |

### 2.17 Ensure 'clr strict security' Server Configuration Option is set to '1'

### (Automated)

**Profile Applicability:**

* Level 1 - Database Engine
* Level 1 - AWS RDS

**Description:**

The clr strict security option specifies whether the engine applies the PERMISSION\_SET on the assemblies.

**Rationale:**

Enabling use of CLR assemblies widens the attack surface of SQL Server and puts it at risk from both inadvertent and malicious assemblies.

**Impact:**

If CLR assemblies are in use, applications may need to be rearchitected to eliminate their usage before enabling this setting. To find user-created assemblies, run the following query in all databases, replacing <database\_name> with each database name:

USE [<database\_name>]

GO

SELECT name AS Assembly\_Name, permission\_set\_desc

FROM sys.assemblies

WHERE is\_user\_defined = 1;

GO

**Audit:**

Run the following T-SQL command:

SELECT name,

CAST(value as int) as value\_configured,

CAST(value\_in\_use as int) as value\_in\_use

FROM sys.configurations

WHERE name = 'clr strict security';

Both value columns must show 1 to be compliant.

**Remediation:**

Run the following T-SQL command:

EXECUTE sp\_configure 'show advanced options', 1;

RECONFIGURE;

EXECUTE sp\_configure 'clr strict security', 1;

RECONFIGURE;

GO

EXECUTE sp\_configure 'show advanced options', 0; RECONFIGURE;

**Default Value:**

By default, this option is Enabled (1).

**References:**

1. https://docs.microsoft.com/en-us/sql/database-engine/configure-windows/clrstrict-security?view=sql-server-ver15&viewFallbackFrom=sql-server-2019

**CIS Controls:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Controls Version** | **Control** | **IG 1** | **IG 2** | **IG 3** |
| v7 | 18.9 Separate Production and Non-Production Systems  Maintain separate environments for production and nonproduction systems.  Developers should not have unmonitored access to production environments. |  | ● | ● |
| v7 | 18.11 Use Standard Hardening Configuration Templates for Databases  For applications that rely on a database, use standard hardening configuration templates. All systems that are part of critical business processes should also be tested. |  | ● | ● |

# *3 Authentication and Authorization*

This section contains recommendations related to SQL Server's authentication and authorization mechanisms.

### 3.1 Ensure 'Server Authentication' Property is set to 'Windows

### Authentication Mode' (Automated)

**Profile Applicability:**

* Level 1 - Database Engine
* Level 1 - AWS RDS

**Description:**

Uses **Windows Authentication** to validate attempted connections.

**Rationale:**

Windows provides a more robust authentication mechanism than SQL Server authentication.

**Impact:**

Changing the login mode configuration requires a restart of the service.

**Audit:**

Execute the following syntax:

SELECT SERVERPROPERTY('IsIntegratedSecurityOnly') as [login\_mode];

A login\_mode of 1 indicates the **Server Authentication** property is set to **Windows Authentication Mode**. A login\_mode of 0 indicates mixed mode authentication.

**Remediation:**

Perform either the GUI or T-SQL method shown:

***1.1.1.5 GUI Method***

1. Open **SQL Server Management Studio**.
2. Open the **Object Explorer** tab and connect to the target database instance.
3. Right click the instance name and select **Properties**.
4. Select the **Security** page from the left menu.
5. Set the **Server authentication** setting to **Windows Authentication Mode**.

***1.1.1.6 T-SQL Method***

Run the following T-SQL in a Query Window:

USE [master]

GO

EXEC xp\_instance\_regwrite N'HKEY\_LOCAL\_MACHINE',

N'Software\Microsoft\MSSQLServer\MSSQLServer', N'LoginMode', REG\_DWORD, 1 GO

Restart the SQL Server service for the change to take effect.

**Default Value:**

Windows Authentication Mode

**References:**

1. https://docs.microsoft.com/en-us/sql/database-engine/configurewindows/server-properties-security-page

**CIS Controls:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Controls Version** | **Control** | **IG 1** | **IG 2** | **IG 3** |
| v7 | 16.2 Configure Centralized Point of Authentication  Configure access for all accounts through as few centralized points of authentication as possible, including network, security, and cloud systems. |  | ● | ● |
| v6 | 16.9 Configure Account Access Centrally  Configure access for all accounts through a centralized point of authentication, for example Active Directory or LDAP. Configure network and security devices for centralized authentication as well. |  |  |  |

### 3.2 Ensure CONNECT permissions on the 'guest' user is Revoked within all SQL Server databases excluding the master, msdb and tempdb (Automated)

**Profile Applicability:**

* Level 1 - Database Engine
* Level 1 - AWS RDS

**Description:**

Remove the right of the guest user to connect to SQL Server databases, except for master, msdb, and tempdb.

**Rationale:**

A login assumes the identity of the guest user when a login has access to SQL Server but does not have access to a database through its own account and the database has a guest user account. Revoking the CONNECT permission for the guest user will ensure that a login is not able to access database information without explicit access to do so.

**Impact:**

When CONNECT permission to the guest user is revoked, a SQL Server instance login must be mapped to a database user explicitly in order to have access to the database.

**Audit:**

Run the following code snippet for each database (replacing *<database\_name>* as appropriate) in the instance to determine if the guest user has CONNECT permission. No rows should be returned.

USE <database\_name>;

GO

SELECT DB\_NAME() AS DatabaseName, 'guest' AS Database\_User,

[permission\_name], [state\_desc]

FROM sys.database\_permissions

WHERE [grantee\_principal\_id] = DATABASE\_PRINCIPAL\_ID('guest')

AND [state\_desc] LIKE 'GRANT%'

AND [permission\_name] = 'CONNECT'

AND DB\_NAME() NOT IN ('master','tempdb','msdb');

**Remediation:**

The following code snippet revokes CONNECT permissions from the guest user in a database. Replace *<database\_name>* as appropriate:

USE <database\_name>;

GO

REVOKE CONNECT FROM guest;

**Default Value:**

The guest user account is added to each new database but without CONNECT permission by default.

**References:**

1. https://docs.microsoft.com/en-us/sql/relational-databases/policy-basedmanagement/guest-permissions-on-user-databases

**Additional Information:**

The guest user cannot have the CONNECT permission revoked in master, msdb and tempdb, but this permission should be revoked in all other databases on the SQL Server instance.

**CIS Controls:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Controls Version** | **Control** | **IG 1** | **IG 2** | **IG 3** |
| v7 | 14.6 Protect Information through Access Control Lists  Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities. | ● | ● | ● |
| v6 | 16 Account Monitoring and Control  Account Monitoring and Control |  |  |  |

### 3.3 Ensure 'Orphaned Users' are Dropped From SQL Server Databases

### (Automated)

**Profile Applicability:**

* Level 1 - Database Engine
* Level 1 - AWS RDS

**Description:**

A database user for which the corresponding SQL Server login is undefined or is incorrectly defined on a server instance cannot log in to the instance and is referred to as orphaned and should be removed.

**Rationale:**

Orphan users should be removed to avoid potential misuse of those broken users in any way.

**Audit:**

Run the following T-SQL query in each database to identify orphan users. No rows should be returned.

USE <database\_name>;

GO

EXEC sp\_change\_users\_login @Action='Report';

**Remediation:**

If the orphaned user cannot or should not be matched to an existing or new login using the Microsoft documented process referenced below, run the following T-SQL query in the appropriate database to remove an orphan user:

USE <database\_name>;

GO

DROP USER <username>;

**References:**

1. https://docs.microsoft.com/en-us/sql/sql-server/failover-clusters/troubleshootorphaned-users-sql-server

**CIS Controls:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Controls Version** | **Control** | **IG 1** | **IG 2** | **IG 3** |
| v7 | 16.8 Disable Any Unassociated Accounts  Disable any account that cannot be associated with a business process or business owner. | ● | ● | ● |
| v6 | 16 Account Monitoring and Control  Account Monitoring and Control |  |  |  |

### 3.4 Ensure SQL Authentication is not used in contained databases (Automated)

**Profile Applicability:**

* Level 1 - Database Engine
* Level 1 - AWS RDS

**Description:**

Contained databases do not enforce password complexity rules for SQL Authenticated users.

**Rationale:**

The absence of an enforced password policy may increase the likelihood of a weak credential being established in a contained database.

**Impact:**

While contained databases provide flexibility in relocating databases to different instances and different environments, this must be balanced with the consideration that no password policy mechanism exists for SQL Authenticated users in contained databases.

**Audit:**

Execute the following T-SQL in each contained database to find database users that are using SQL authentication:

|  |
| --- |
| SELECT name AS DBUser  FROM sys.database\_principals  WHERE name NOT IN ('dbo','Information\_Schema','sys','guest')  AND type IN ('U','S','G')  AND authentication\_type = 2; GO |

**Remediation:**

Leverage Windows Authenticated users in contained databases.

**Default Value:**

SQL Authenticated users (USER WITH PASSWORD authentication) are allowed in contained databases.

**References:**

1. https://docs.microsoft.com/en-us/sql/relational-databases/databases/securitybest-practices-with-contained-databases

**CIS Controls:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Controls Version** | **Control** | **IG 1** | **IG 2** | **IG 3** |
| v7 | 16.2 Configure Centralized Point of Authentication  Configure access for all accounts through as few centralized points of authentication as possible, including network, security, and cloud systems. |  | ● | ● |
| v6 | 16.12 Use Long Passwords For All User Accounts  Where multi-factor authentication is not supported, user accounts shall be required to use long passwords on the system (longer than 14 characters). |  |  |  |

### 3.5 Ensure the SQL Server’s MSSQL Service Account is Not an

### Administrator (Manual)

**Profile Applicability:**

* Level 1 - Database Engine
* Level 1 - AWS RDS

**Description:**

The service account and/or service SID used by the MSSQLSERVER service for a default instance or *<InstanceName>* service for a named instance should not be a member of the Windows Administrator group either directly or indirectly (via a group). This also means that the account known as LocalSystem (aka NT AUTHORITY\SYSTEM) should not be used for the MSSQL service as this account has higher privileges than the SQL Server service requires.

**Rationale:**

Following the principle of least privilege, the service account should have no more privileges than required to do its job. For SQL Server services, the SQL Server Setup will assign the required permissions directly to the service SID. No additional permissions or privileges should be necessary.

**Impact:**

The **SQL Server Configuration Manager** tool should always be used to change the SQL Server’s service account. This will ensure that the account has the necessary privileges. If the service needs access to resources other than the standard Microsoft defined directories and registry, then additional permissions may need to be granted separately to those resources.

**Audit:**

Verify that the service account (in case of a local or AD account) and service SID are not members of the Windows Administrators group.

**Remediation:**

In the case where LocalSystem is used, use **SQL Server Configuration Manager** to change to a less privileged account. Otherwise, remove the account or service SID from the Administrators group. You may need to run the **SQL Server Configuration Manager** if underlying permissions had been changed or if **SQL Server Configuration Manager** was not originally used to set the service account.

**Default Value:**

By default, the Service Account (or Service SID) is not a member of the Administrators group.

**References:**

1. https://docs.microsoft.com/en-us/sql/database-engine/configurewindows/configure-windows-service-accounts-and-permissions

**CIS Controls:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Controls Version** | **Control** | **IG 1** | **IG 2** | **IG 3** |
| v7 | 4.3 Ensure the Use of Dedicated Administrative Accounts  Ensure that all users with administrative account access use a dedicated or secondary account for elevated activities. This account should only be used for administrative activities and not internet browsing, email, or similar activities. | ● | ● | ● |
| v6 | 5.1 Minimize And Sparingly Use Administrative Privileges  Minimize administrative privileges and only use administrative accounts when they are required. Implement focused auditing on the use of administrative privileged functions and monitor for anomalous behavior. |  |  |  |

### 3.6 Ensure the SQL Server’s SQLAgent Service Account is Not an

### Administrator (Manual)

**Profile Applicability:**

* Level 1 - Database Engine
* Level 1 - AWS RDS

**Description:**

The service account and/or service SID used by the SQLSERVERAGENT service for a default instance or SQLAGENT$*<InstanceName>* service for a named instance should not be a member of the Windows Administrator group either directly or indirectly (via a group). This also means that the account known as LocalSystem (AKA NT AUTHORITY\SYSTEM) should not be used for the SQLAGENT service as this account has higher privileges than the SQL Server service requires.

**Rationale:**

Following the principle of least privilege, the service account should have no more privileges than required to do its job. For SQL Server services, the SQL Server Setup will assign the required permissions directly to the service SID. No additional permissions or privileges should be necessary.

**Impact:**

The **SQL Server Configuration Manager** tool should always be used to change the SQL Server’s service account. This will ensure that the account has the necessary privileges. If the service needs access to resources other than the standard Microsoft-defined directories and registry, then additional permissions may need to be granted separately to those resources.

If using the auto restart feature, then the SQLAGENT service must be an Administrator.

**Audit:**

Verify that the service account (in case of a local or AD account) and service SID are not members of the Windows Administrators group.

**Remediation:**

In the case where LocalSystem is used, use **SQL Server Configuration Manager** to change to a less privileged account. Otherwise, remove the account or service SID from the Administrators group. You may need to run the **SQL Server Configuration Manager** if underlying permissions had been changed or if **SQL Server Configuration Manager** was not originally used to set the service account.

**Default Value:**

By default, the Service Account (or Service SID) is not a member of the Administrators group.

**References:**

1. https://docs.microsoft.com/en-us/sql/database-engine/configurewindows/configure-windows-service-accounts-and-permissions

**CIS Controls:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Controls Version** | **Control** | **IG 1** | **IG 2** | **IG 3** |
| v7 | 4.3 Ensure the Use of Dedicated Administrative Accounts  Ensure that all users with administrative account access use a dedicated or secondary account for elevated activities. This account should only be used for administrative activities and not internet browsing, email, or similar activities. | ● | ● | ● |
| v6 | 5.1 Minimize And Sparingly Use Administrative Privileges  Minimize administrative privileges and only use administrative accounts when they are required. Implement focused auditing on the use of administrative privileged functions and monitor for anomalous behavior. |  |  |  |

### 3.7 Ensure the SQL Server’s Full-Text Service Account is Not an

### Administrator (Manual)

**Profile Applicability:**

* Level 1 - Database Engine
* Level 1 - AWS RDS

**Description:**

The service account and/or service SID used by the MSSQLFDLauncher service for a default instance or MSSQLFDLauncher$*<InstanceName>* service for a named instance should not be a member of the Windows Administrator group either directly or indirectly (via a group). This also means that the account known as LocalSystem (aka NT AUTHORITY\SYSTEM) should not be used for the Full-Text service as this account has higher privileges than the SQL Server service requires.

**Rationale:**

Following the principle of least privilege, the service account should have no more privileges than required to do its job. For SQL Server services, the SQL Server Setup will assign the required permissions directly to the service SID. No additional permissions or privileges should be necessary.

**Impact:**

The **SQL Server Configuration Manager** tool should always be used to change the SQL Server’s service account. This will ensure that the account has the necessary privileges. If the service needs access to resources other than the standard Microsoft-defined directories and registry, then additional permissions may need to be granted separately to those resources.

**Audit:**

Verify that the service account (in case of a local or AD account) and service SID are not members of the Windows Administrators group.

**Remediation:**

In the case where LocalSystem is used, use **SQL Server Configuration Manager** to change to a less privileged account. Otherwise, remove the account or service SID from the Administrators group. You may need to run the **SQL Server Configuration Manager** if underlying permissions had been changed or if **SQL Server Configuration Manager** was not originally used to set the service account.

**Default Value:**

By default, the Service Account (or Service SID) is not a member of the Administrators group.

**References:**

1. https://docs.microsoft.com/en-us/sql/database-engine/configurewindows/configure-windows-service-accounts-and-permissions

**CIS Controls:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Controls Version** | **Control** | **IG 1** | **IG 2** | **IG 3** |
| v7 | 4.3 Ensure the Use of Dedicated Administrative Accounts  Ensure that all users with administrative account access use a dedicated or secondary account for elevated activities. This account should only be used for administrative activities and not internet browsing, email, or similar activities. | ● | ● | ● |
| v6 | 5.1 Minimize And Sparingly Use Administrative Privileges  Minimize administrative privileges and only use administrative accounts when they are required. Implement focused auditing on the use of administrative privileged functions and monitor for anomalous behavior. |  |  |  |

### 3.8 Ensure only the default permissions specified by Microsoft are granted to the public server role (Automated)

**Profile Applicability:**

* Level 1 - Database Engine
* Level 1 - AWS RDS

**Description:**

public is a special fixed server role containing all logins. Unlike other fixed server roles, permissions can be changed for the public role. In keeping with the principle of least privileges, the public server role should not be used to grant permissions at the server scope as these would be inherited by all users.

**Rationale:**

Every SQL Server login belongs to the public role and cannot be removed from this role. Therefore, any permissions granted to this role will be available to all logins unless they have been explicitly denied to specific logins or user-defined server roles.

**Impact:**

When the extraneous permissions are revoked from the public server role, access may be lost unless the permissions are granted to the explicit logins or to user-defined server roles containing the logins which require the access.

**Audit:**

Use the following syntax to determine if extra permissions have been granted to the public server role.

SELECT \*

FROM master.sys.server\_permissions

WHERE (grantee\_principal\_id = SUSER\_SID(N'public') and state\_desc LIKE

'GRANT%')

AND NOT (state\_desc = 'GRANT' and [permission\_name] = 'VIEW ANY DATABASE' and class\_desc = 'SERVER')

AND NOT (state\_desc = 'GRANT' and [permission\_name] = 'CONNECT' and class\_desc = 'ENDPOINT' and major\_id = 2)

AND NOT (state\_desc = 'GRANT' and [permission\_name] = 'CONNECT' and class\_desc = 'ENDPOINT' and major\_id = 3)

AND NOT (state\_desc = 'GRANT' and [permission\_name] = 'CONNECT' and class\_desc = 'ENDPOINT' and major\_id = 4)

AND NOT (state\_desc = 'GRANT' and [permission\_name] = 'CONNECT' and class\_desc = 'ENDPOINT' and major\_id = 5);

This query should not return any rows.

**Remediation:**

1. Add the extraneous permissions found in the Audit query results to the specific logins to user-defined server roles which require the access.
2. Revoke the *<permission\_name>* from the public role as shown below

USE [master]

GO

REVOKE <permission\_name> FROM public; GO

**Default Value:**

By default, the public server role is granted VIEW ANY DATABASE permission and the

CONNECT permission on the default endpoints (TSQL Local Machine, TSQL Named Pipes, TSQL Default TCP, TSQL Default VIA). The VIEW ANY DATABASE permission allows all logins to see database metadata, unless explicitly denied. **References:**

1. https://docs.microsoft.com/en-us/sql/relationaldatabases/security/authentication-access/server-level-roles
2. https://docs.microsoft.com/en-us/sql/relationaldatabases/security/authentication-access/server-level-roles#permissions-of-fixedserver-roles

**CIS Controls:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Controls Version** | **Control** | **IG 1** | **IG 2** | **IG 3** |
| v7 | 14.6 Protect Information through Access Control Lists  Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities. | ● | ● | ● |
| v6 | 5.1 Minimize And Sparingly Use Administrative Privileges  Minimize administrative privileges and only use administrative accounts when they are required. Implement focused auditing on the use of administrative privileged functions and monitor for anomalous behavior. |  |  |  |

### 3.9 Ensure Windows BUILTIN groups are not SQL Logins (Automated)

**Profile Applicability:**

* Level 1 - Database Engine
* Level 1 - AWS RDS

**Description:**

Prior to SQL Server 2008, the BUILTIN\Administrators group was added as a SQL Server login with sysadmin privileges during installation by default. Best practices promote creating an Active Directory level group containing approved DBA staff accounts and using this controlled AD group as the login with sysadmin privileges. The AD group should be specified during SQL Server installation and the BUILTIN\Administrators group would therefore have no need to be a login.

**Rationale:**

The BUILTIN groups (Administrators, Everyone, Authenticated Users, Guests, etc.) generally contain very broad memberships which would not meet the best practice of ensuring only the necessary users have been granted access to a SQL Server instance. These groups should not be used for any level of access into a SQL Server Database Engine instance.

**Impact:**

Before dropping the BUILTIN group logins, ensure that alternative AD Groups or Windows logins have been added with equivalent permissions. Otherwise, the SQL Server instance may become totally inaccessible.

**Audit:**

Use the following syntax to determine if any BUILTIN groups or accounts have been added as SQL Server Logins.

SELECT pr.[name], pe.[permission\_name], pe.[state\_desc]

FROM sys.server\_principals pr

JOIN sys.server\_permissions pe

ON pr.principal\_id = pe.grantee\_principal\_id WHERE pr.name like 'BUILTIN%';

This query should not return any rows.

**Remediation:**

1. For each BUILTIN login, if needed create a more restrictive AD group containing only the required user accounts.
2. Add the AD group or individual Windows accounts as a SQL Server login and grant it the permissions required.
3. Drop the BUILTIN login using the syntax below after replacing *<name>* in [BUILTIN\*<name>*].

USE [master]

GO

DROP LOGIN [BUILTIN\<name>] GO

**Default Value:**

By default, no BUILTIN groups are added as SQL logins.

**CIS Controls:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Controls Version** | **Control** | **IG 1** | **IG 2** | **IG 3** |
| v7 | 14.6 Protect Information through Access Control Lists  Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities. | ● | ● | ● |
| v6 | 14.4 Protect Information With Access Control Lists  All information stored on systems shall be protected with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities. |  |  |  |

### 3.10 Ensure Windows local groups are not SQL Logins (Automated)

**Profile Applicability:**

* Level 1 - Database Engine
* Level 1 - AWS RDS

**Description:**

Local Windows groups should not be used as logins for SQL Server instances.

**Rationale:**

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.

**Impact:**

Before dropping the local group logins, ensure that alternative AD Groups or Windows logins have been added with equivalent permissions. Otherwise, the SQL Server instance may become totally inaccessible.

**Audit:**

Use the following syntax to determine if any local groups have been added as SQL Server Logins.

|  |
| --- |
| USE [master] GO  SELECT pr.[name] AS LocalGroupName, pe.[permission\_name], pe.[state\_desc]  FROM sys.server\_principals pr  JOIN sys.server\_permissions pe  ON pr.[principal\_id] = pe.[grantee\_principal\_id]  WHERE pr.[type\_desc] = 'WINDOWS\_GROUP'  AND pr.[name] like CAST(SERVERPROPERTY('MachineName') AS nvarchar) + '%'; |

This query should not return any rows.

**Remediation:**

1. For each LocalGroupName login, if needed create an equivalent AD group containing only the required user accounts.
2. Add the AD group or individual Windows accounts as a SQL Server login and grant it the permissions required.
3. Drop the LocalGroupName login using the syntax below after replacing *<name>*.

USE [master]

GO

DROP LOGIN [<name>] GO

**Default Value:**

By default, no local groups are added as SQL logins.

**CIS Controls:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Controls Version** | **Control** | **IG 1** | **IG 2** | **IG 3** |
| v7 | 14.6 Protect Information through Access Control Lists  Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities. | ● | ● | ● |
| v6 | 14.4 Protect Information With Access Control Lists  All information stored on systems shall be protected with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities. |  |  |  |

### 3.11 Ensure the public role in the msdb database is not granted access to SQL Agent proxies (Automated)

**Profile Applicability:**

• Level 1 - Database Engine

**Description:**

The public database role contains every user in the msdb database. SQL Agent proxies define a security context in which a job step can run.

**Rationale:**

Granting access to SQL Agent proxies for the public role would allow all users to utilize the proxy which may have high privileges. This would likely break the principle of least privileges.

**Impact:**

Before revoking the public role from the proxy, ensure that alternative logins or appropriate user-defined database roles have been added with equivalent permissions.

Otherwise, SQL Agent job steps dependent upon this access will fail.

**Audit:**

Use the following syntax to determine if access to any proxies have been granted to the msdb database's public role.

USE [msdb]

GO

SELECT sp.name AS proxyname

FROM dbo.sysproxylogin spl

JOIN sys.database\_principals dp

ON dp.sid = spl.sid

JOIN sysproxies sp

ON sp.proxy\_id = spl.proxy\_id

WHERE principal\_id = USER\_ID('public'); GO

This query should not return any rows.

**Remediation:**

1. Ensure the required security principals are explicitly granted access to the proxy (use sp\_grant\_login\_to\_proxy).
2. Revoke access to the *<proxyname>* from the public role.

|  |
| --- |
| USE [msdb] GO  EXEC dbo.sp\_revoke\_login\_from\_proxy @name = N'public', @proxy\_name = N'<proxyname>';  GO |

**Default Value:**

By default, the msdb public database role does not have access to any proxy.

**References:**

1. https://support.microsoft.com/en-us/help/2160741/best-practices-in-configuringsql-server-agent-proxy-account

**CIS Controls:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Controls Version** | **Control** | **IG 1** | **IG 2** | **IG 3** |
| v7 | 14.6 Protect Information through Access Control Lists  Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities. | ● | ● | ● |
| v6 | 14.4 Protect Information With Access Control Lists  All information stored on systems shall be protected with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities. |  |  |  |

# *4 Password Policies*

This section contains recommendations related to SQL Server's password policies.

### 4.1 Ensure 'MUST\_CHANGE' Option is set to 'ON' for All SQL Authenticated Logins (Manual)

**Profile Applicability:**

• Level 1 - Database Engine

**Description:**

Whenever this option is set to ON, SQL Server will prompt for an updated password the first time the new or altered login is used.

**Rationale:**

Enforcing a password change after a reset or new login creation will prevent the account administrators or anyone accessing the initial password from misuse of the SQL login created without being noticed.

**Impact:**

CHECK\_EXPIRATION and CHECK\_POLICY options must both be ON. End users must have the means (application) to change the password when forced.

**Audit:**

1. Open **SQL Server Management Studio**.
2. Open **Object Explorer** and connect to the target instance.
3. Navigate to the **Logins** tab in **Object Explorer** and expand. Right click on the desired login and select **Properties**.
4. Verify the User must change password at next login checkbox is checked.

**Note:** This audit procedure is only applicable immediately after the login has been created or altered to force the password change. Once the password is changed, there is no way to know specifically that this option was the forcing mechanism behind a password change.

**Remediation:**

Set the MUST\_CHANGE option for SQL Authenticated logins when creating a login initially:

CREATE LOGIN <login\_name> WITH PASSWORD = '<password\_value>' MUST\_CHANGE, CHECK\_EXPIRATION = ON, CHECK\_POLICY = ON;

Set the MUST\_CHANGE option for SQL Authenticated logins when resetting a password:

ALTER LOGIN <login\_name> WITH PASSWORD = '<new\_password\_value>' MUST\_CHANGE;

**Default Value:**

ON when creating a new login via the SSMS GUI. OFF when creating a new login using T-SQL CREATE LOGIN unless the MUST\_CHANGE option is explicitly included along with CHECK\_EXPIRATION = ON.

**References:**

1. https://docs.microsoft.com/en-us/sql/t-sql/statements/alter-login-transact-sql
2. https://docs.microsoft.com/en-us/sql/t-sql/statements/create-login-transact-sql

**CIS Controls:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Controls Version** | **Control** | **IG 1** | **IG 2** | **IG 3** |
| v7 | 4.2 Change Default Passwords  Before deploying any new asset, change all default passwords to have values consistent with administrative level accounts. | ● | ● | ● |
| v6 | 16 Account Monitoring and Control  Account Monitoring and Control |  |  |  |

#### 4.2 Ensure 'CHECK\_EXPIRATION' Option is set to 'ON' for All SQL

#### Authenticated Logins Within the Sysadmin Role (Automated)

**Profile Applicability:**

* Level 1 - Database Engine
* Level 1 - AWS RDS

**Description:**

Applies the same password expiration policy used in Windows to passwords used inside SQL Server.

**Rationale:**

Ensuring SQL logins comply with the secure password policy applied by the Windows Server Benchmark will ensure the passwords for SQL logins with sysadmin privileges are changed on a frequent basis to help prevent compromise via a brute force attack. CONTROL SERVER is an equivalent permission to sysadmin and logins with that permission should also be required to have expiring passwords.

**Impact:**

This is a mitigating recommendation for systems which cannot follow the recommendation to use only Windows Authenticated logins.

Regarding limiting this rule to only logins with sysadmin and CONTROL SERVER privileges, there are too many cases of applications that run with less than sysadmin level privileges that have hard-coded passwords or effectively hard-coded passwords (whatever is set the first time is nearly impossible to change). There are several line-of-business applications that are considered best of breed which have this failing.

Also, keep in mind that the password policy is taken from the computer's local policy, which is taken from the Default Domain Policy setting. Many organizations have a different password policy regarding the service accounts. These are handled in AD by setting the account's password to not expire and having some other process track when the password needs to be changed. With this second control in place, this is perfectly acceptable from an audit perspective. If you treat a SQL Server login as a service account, then you have to do the same. This ensures that the password change happens during a communicated downtime window and not arbitrarily.

**Audit:**

Run the following T-SQL statement to find sysadmin or equivalent logins with CHECK\_EXPIRATION = OFF. No rows should be returned.

SELECT l.[name], 'sysadmin membership' AS 'Access\_Method'

FROM sys.sql\_logins AS l

WHERE IS\_SRVROLEMEMBER('sysadmin',name) = 1

AND l.is\_expiration\_checked <> 1

UNION ALL

SELECT l.[name], 'CONTROL SERVER' AS 'Access\_Method'

FROM sys.sql\_logins AS l

JOIN sys.server\_permissions AS p

ON l.principal\_id = p.grantee\_principal\_id

WHERE p.type = 'CL' AND p.state IN ('G', 'W') AND l.is\_expiration\_checked <> 1;

**Remediation:**

For each *<login\_name>* found by the Audit Procedure, execute the following T-SQL statement:

ALTER LOGIN [<login\_name>] WITH CHECK\_EXPIRATION = ON;

**Default Value:**

CHECK\_EXPIRATION is ON by default when using SSMS to create a SQL authenticated login.

CHECK\_EXPIRATION is OFF by default when using T-SQL CREATE LOGIN syntax without specifying the CHECK\_EXPIRATION option.

**References:**

1. https://docs.microsoft.com/en-us/sql/relational-databases/security/passwordpolicy?view=sql-server-ver15

**CIS Controls:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Controls Version** | **Control** | **IG 1** | **IG 2** | **IG 3** |
| v7 | 16.10 Ensure All Accounts Have An Expiration Date  Ensure that all accounts have an expiration date that is monitored and enforced. |  | ● | ● |
| v6 | 16.2 All Accounts Have A Monitored Expiration Date  Ensure that all accounts have an expiration date that is monitored and enforced. |  |  |  |

#### 4.3 Ensure 'CHECK\_POLICY' Option is set to 'ON' for All SQL Authenticated Logins (Automated)

**Profile Applicability:**

* Level 1 - Database Engine
* Level 1 - AWS RDS

**Description:**

Applies the same password complexity policy used in Windows to passwords used inside SQL Server.

**Rationale:**

Ensure SQL authenticated login passwords comply with the secure password policy applied by the Windows Server Benchmark so that they cannot be easily compromised via brute force attack.

**Impact:**

This is a mitigating recommendation for systems which cannot follow the recommendation to use only Windows Authenticated logins.

Weak passwords can lead to compromised systems. SQL Server authenticated logins will utilize the password policy set in the computer's local policy, which is typically set by the Default Domain Policy setting.

The setting is only enforced when the password is changed. This setting does not force existing weak passwords to be changed.

**Audit:**

Use the following code snippet to determine the status of SQL Logins and if their password complexity is enforced.

SELECT name, is\_disabled

FROM sys.sql\_logins

WHERE is\_policy\_checked = 0;

The is\_policy\_checked value of 0 indicates that the CHECK\_POLICY option is OFF; value of 1 is ON. If is\_disabled value is 1, then the login is disabled and unusable. If no rows are returned then either no SQL Authenticated logins exist or they all have CHECK\_POLICY ON.

**Remediation:**

For each *<login\_name>* found by the Audit Procedure, execute the following T-SQL statement:

ALTER LOGIN [<login\_name>] WITH CHECK\_POLICY = ON;

**Note:** In the case of AWS RDS do not perform this remediation for the Master account.

**Default Value:**

CHECK\_POLICY is ON **References:**

1. https://docs.microsoft.com/en-us/sql/relational-databases/security/passwordpolicy

**CIS Controls:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Controls Version** | **Control** | **IG 1** | **IG 2** | **IG 3** |
| v7 | 4.4 Use Unique Passwords  Where multi-factor authentication is not supported (such as local administrator, root, or service accounts), accounts will use passwords that are unique to that system. |  | ● | ● |
| v6 | 16 Account Monitoring and Control  Account Monitoring and Control |  |  |  |

## 5 Auditing and Logging

This section contains recommendations related to SQL Server's audit and logging mechanisms.

#### 5.1 Ensure 'Maximum number of error log files' is set to greater than or equal to '12' (Automated)

**Profile Applicability:**

* Level 1 - Database Engine
* Level 1 - AWS RDS

**Description:**

SQL Server error log files must be protected from loss. The log files must be backed up before they are overwritten. Retaining more error logs helps prevent loss from frequent recycling before backups can occur.

**Rationale:**

The SQL Server error log contains important information about major server events and login attempt information as well.

**Impact:**

Once the max number of error logs is reached, the oldest error log file is deleted each time SQL Server restarts or sp\_cycle\_errorlog is executed.

**Audit:**

Perform either the GUI or T-SQL method shown:

***1.1.1.7 GUI Method***

1. Open **SQL Server Management Studio**.
2. Open **Object Explorer** and connect to the target instance.
3. Navigate to the **Management** tab in **Object Explorer** and expand. Right click on the **SQL Server Logs** file and select **Configure**.
4. Verify the **Limit the number of error log files before they are recycled** checkbox is checked
5. Verify the **Maximum number of error log files** is greater than or equal to 12

***1.1.1.8 T-SQL Method***

Run the following T-SQL. The NumberOfLogFiles returned should be greater than or equal to 12.

DECLARE @NumErrorLogs int;

EXEC master.sys.xp\_instance\_regread

N'HKEY\_LOCAL\_MACHINE',

N'Software\Microsoft\MSSQLServer\MSSQLServer',

N'NumErrorLogs',

@NumErrorLogs OUTPUT;

SELECT ISNULL(@NumErrorLogs, -1) AS [NumberOfLogFiles];

**Remediation:**

Adjust the number of logs to prevent data loss. The default value of 6 may be insufficient for a production environment. Perform either the GUI or T-SQL method shown:

***1.1.1.9 GUI Method***

1. Open **SQL Server Management Studio**.
2. Open **Object Explorer** and connect to the target instance.
3. Navigate to the **Management** tab in **Object Explorer** and expand. Right click on the **SQL Server Logs** file and select **Configure**
4. Check the **Limit the number of error log files before they are recycled**
5. Set the **Maximum number of error log files** to greater than or equal to 12

***1.1.1.10 T-SQL Method***

Run the following T-SQL to change the number of error log files, replace *<NumberAbove12>* with your desired number of error log files:

EXEC master.sys.xp\_instance\_regwrite

N'HKEY\_LOCAL\_MACHINE',

N'Software\Microsoft\MSSQLServer\MSSQLServer',

N'NumErrorLogs',

REG\_DWORD, <NumberAbove12>;

**Default Value:**

6 SQL Server error log files in addition to the current error log file are retained by default.

**References:**

1. https://docs.microsoft.com/en-us/sql/database-engine/configure-windows/scmservices-configure-sql-server-error-logs

**CIS Controls:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Controls Version** | **Control** | **IG 1** | **IG 2** | **IG 3** |
| v7 | 6.4 Ensure adequate storage for logs  Ensure that all systems that store logs have adequate storage space for the logs generated. |  | ● | ● |
| v6 | 6.3 Ensure Audit Logging Systems Are Not Subject To Loss (i.e. rotation/archive)  Ensure that all systems that store logs have adequate storage space for the logs generated on a regular basis, so that log files will not fill up between log rotation intervals. The logs must be archived and digitally signed on a periodic basis. |  |  |  |

#### 5.2 Ensure 'Default Trace Enabled' Server Configuration Option is set to

#### '1' (Automated)

**Profile Applicability:**

* Level 1 - Database Engine
* Level 1 - AWS RDS

**Description:**

The default trace provides audit logging of database activity including account creations, privilege elevation and execution of DBCC commands.

**Rationale:**

Default trace provides valuable audit information regarding security-related activities on the server.

**Audit:**

Run the following T-SQL command:

SELECT name,

CAST(value as int) as value\_configured,

CAST(value\_in\_use as int) as value\_in\_use

FROM sys.configurations

WHERE name = 'default trace enabled';

Both value columns must show 1.

**Remediation:**

Run the following T-SQL command:

EXECUTE sp\_configure 'show advanced options', 1;

RECONFIGURE;

EXECUTE sp\_configure 'default trace enabled', 1;

RECONFIGURE;

GO

EXECUTE sp\_configure 'show advanced options', 0; RECONFIGURE;

**Default Value:**

1 (on)

**References:**

1. https://docs.microsoft.com/en-us/sql/database-engine/configurewindows/default-trace-enabled-server-configuration-option

**CIS Controls:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Controls Version** | **Control** | **IG 1** | **IG 2** | **IG 3** |
| v7 | 6.2 Activate audit logging  Ensure that local logging has been enabled on all systems and networking devices. | ● | ● | ● |
| v7 | 6.3 Enable Detailed Logging  Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements. |  | ● | ● |
| v6 | 6.2 Ensure Audit Log Settings Support Appropriate Log Entry Formatting  Validate audit log settings for each hardware device and the software installed on it, ensuring that logs include a date, timestamp, source addresses, destination addresses, and various other useful elements of each packet and/or transaction. Systems should record logs in a standardized format such as syslog entries or those outlined by the Common Event Expression initiative. If systems cannot generate logs in a standardized format, log normalization tools can be deployed to convert logs into such a format. |  |  |  |

#### 5.3 Ensure 'Login Auditing' is set to 'failed logins' (Automated)

**Profile Applicability:**

* Level 1 - Database Engine
* Level 1 - AWS RDS

**Description:**

This setting will record failed authentication attempts for SQL Server logins to the **SQL Server Errorlog**. This is the default setting for SQL Server.

Historically, this setting has been available in all versions and editions of SQL Server. Prior to the availability of **SQL Server Audit**, this was the only provided mechanism for capturing logins (successful or failed).

**Rationale:**

Capturing failed logins provides key information that can be used to detect\confirm password guessing attacks. Capturing successful login attempts can be used to confirm server access during forensic investigations, but using this audit level setting to also capture successful logins creates excessive noise in the **SQL Server Errorlog** which can hamper a DBA trying to troubleshoot problems. Elsewhere in this benchmark, we recommend using the newer lightweight SQL Server Audit feature to capture both successful and failed logins.

**Impact:**

At a minimum, we want to ensure failed logins are captured in order to detect if an adversary is attempting to brute force passwords or otherwise attempting to access a SQL Server improperly.

Changing the setting requires a restart of the SQL Server service.

**Audit:**

EXEC xp\_loginconfig 'audit level';

A config\_value of failure indicates a server login auditing setting of **Failed logins only**. If a config\_value of all appears, then both failed and successful logins are being logged. Both settings should also be considered valid, but as mentioned capturing successful logins using this method creates lots of noise in the **SQL Server Errorlog**.

**Remediation:**

Perform either the GUI or T-SQL method shown:

***1.1.1.11 GUI Method***

1. Open **SQL Server Management Studio**.
2. Right click the target instance and select **Properties** and navigate to the **Security** tab.
3. Select the option **Failed logins only** under the **Login Auditing** section and click **OK**.
4. Restart the SQL Server instance.

***1.1.1.12 T-SQL Method***

1. Run:

EXEC xp\_instance\_regwrite N'HKEY\_LOCAL\_MACHINE',

N'Software\Microsoft\MSSQLServer\MSSQLServer', N'AuditLevel', REG\_DWORD, 2

1. Restart the SQL Server instance.

**Default Value:**

By default, only failed login attempts are captured.

**References:**

1. https://docs.microsoft.com/en-us/sql/database-engine/configurewindows/server-properties-security-page

**CIS Controls:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Controls Version** | **Control** | **IG 1** | **IG 2** | **IG 3** |
| v7 | 16.13 Alert on Account Login Behavior Deviation  Alert when users deviate from normal login behavior, such as time-of-day, workstation location and duration. |  |  | ● |
| v6 | 16.10 Profile User Account Usage And Monitor For Anomalies  Profile each user’s typical account usage by determining normal time-of-day access and access duration. Reports should be generated that indicate users who have logged in during unusual hours or have exceeded their normal login duration. This includes |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Controls Version** | **Control** | **IG 1** | **IG 2** | **IG 3** |
|  | flagging the use of the user’s credentials from a computer other than computers on which the user generally works. |  |  |  |

#### 5.4 Ensure 'SQL Server Audit' is set to capture both 'failed' and 'successful logins' (Automated)

**Profile Applicability:**

* Level 1 - Database Engine
* Level 1 - AWS RDS

**Description:**

SQL Server Audit is capable of capturing both failed and successful logins and writing them to one of three places: the application event log, the security event log, or the file system. We will use it to capture any login attempt to SQL Server, as well as any attempts to change audit policy. This will also serve to be a second source to record failed login attempts.

**Rationale:**

By utilizing Audit instead of the traditional setting under the Security tab to capture successful logins, we reduce the noise in the ERRORLOG. This keeps it smaller and easier to read for DBAs who are attempting to troubleshoot issues with the SQL Server. Also, the Audit object can write to the security event log, though this requires operating system configuration. This gives an additional option for where to store login events, especially in conjunction with an SIEM.

**Impact:**

With the previous recommendation, only failed logins are captured. If the Audit object is not implemented with the appropriate setting, SQL Server will not capture successful logins, which might prove of use for forensics.

**Audit:**

SELECT

S.name AS 'Audit Name'

, CASE S.is\_state\_enabled

WHEN 1 THEN 'Y'

WHEN 0 THEN 'N' END AS 'Audit Enabled'

, S.type\_desc AS 'Write Location'

, SA.name AS 'Audit Specification Name'

, CASE SA.is\_state\_enabled

WHEN 1 THEN 'Y'

WHEN 0 THEN 'N' END AS 'Audit Specification Enabled'

, SAD.audit\_action\_name

, SAD.audited\_result

FROM sys.server\_audit\_specification\_details AS SAD

JOIN sys.server\_audit\_specifications AS SA

ON SAD.server\_specification\_id = SA.server\_specification\_id

JOIN sys.server\_audits AS S

ON SA.audit\_guid = S.audit\_guid

WHERE SAD.audit\_action\_id IN ('CNAU', 'LGFL', 'LGSD');

The result set should contain 3 rows, one for each of the following audit\_action\_names:

•

AUDIT\_CHANGE\_GROUP

•

FAILED\_LOGIN\_GROUP •

SUCCESSFUL\_LOGIN\_GROUP

Both the Audit and Audit specification should be enabled and the audited\_result should include both success and failure.

**Remediation:**

Perform either the GUI or T-SQL method shown:

***1.1.1.13 GUI Method***

1. Expand the **SQL Server** in **Object Explorer**.
2. Expand the **Security Folder**
3. Right-click on the **Audits** folder and choose **New Audit...**
4. Specify a name for the **Server Audit**.
5. Specify the audit destination details and then click **OK** to save the **Server Audit**.
6. Right-click on **Server Audit Specifications** and choose **New Server Audit Specification...**
7. Name the **Server Audit Specification**
8. Select the just created **Server Audit** in the **Audit** drop-down selection.
9. Click the drop-down under **Audit Action Type** and select AUDIT\_CHANGE\_GROUP.
10. Click the new drop-down **Audit Action Type** and select FAILED\_LOGIN\_GROUP.
11. Click the new drop-down under **Audit Action Type** and select SUCCESSFUL\_LOGIN\_GROUP.
12. Click OK to save the **Server Audit Specification**.
13. Right-click on the new **Server Audit Specification** and select **Enable Server Audit Specification**.
14. Right-click on the new **Server Audit** and select **Enable Server Audit**.

***1.1.1.14 T-SQL Method***

Execute code similar to:

CREATE SERVER AUDIT TrackLogins

TO APPLICATION\_LOG;

GO

CREATE SERVER AUDIT SPECIFICATION TrackAllLogins

FOR SERVER AUDIT TrackLogins

ADD (FAILED\_LOGIN\_GROUP),

ADD (SUCCESSFUL\_LOGIN\_GROUP),

ADD (AUDIT\_CHANGE\_GROUP)

WITH (STATE = ON);

GO

ALTER SERVER AUDIT TrackLogins

WITH (STATE = ON);

GO

**Note:** If the write destination for the Audit object is to be the security event log, see the Books Online topic Write SQL Server Audit Events to the Security Log and follow the appropriate steps.

**Default Value:**

By default, there are no audit object tracking login events.

**References:**

1. https://docs.microsoft.com/en-us/sql/relationaldatabases/security/auditing/create-a-server-audit-and-server-audit-specification

**CIS Controls:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Controls Version** | **Control** | **IG 1** | **IG 2** | **IG 3** |
| v7 | 4.9 Log and Alert on Unsuccessful Administrative Account Login  Configure systems to issue a log entry and alert on unsuccessful logins to an administrative account. |  | ● | ● |
| v6 | 5.5 Log Failed Administrative Login Attempts  Configure systems to issue a log entry and alert on any unsuccessful login to an administrative account. |  |  |  |

## 6 Application Development

This section contains recommendations related to developing applications that interface with SQL Server.

#### 6.1 Ensure Database and Application User Input is Sanitized (Manual)

**Profile Applicability:**

• Level 1 - Database Engine

**Description:**

Always validate user input received from a database client or application by testing type, length, format, and range prior to transmitting it to the database server.

**Rationale:**

Sanitizing user input drastically minimizes risk of SQL injection.

**Impact:**

Sanitize user input may require changes to application code or database object syntax. These changes can require applications or databases to be taken temporarily off-line. Any change to TSQL or application code should be thoroughly tested in testing environment before production implementation.

**Audit:**

Check with the application teams to ensure any database interaction is through the use of stored procedures and not dynamic SQL. Revoke any INSERT, UPDATE, or DELETE privileges to users so that modifications to data must be done through stored procedures. Verify that there's no SQL query in the application code produced by string concatenation.

**Remediation:**

The following steps can be taken to remediate SQL injection vulnerabilities:

* Review TSQL and application code for SQL Injection
* Only permit minimally privileged accounts to send user input to the server
* Minimize the risk of SQL injection attack by using parameterized commands and stored procedures
* Reject user input containing binary data, escape sequences, and comment characters
* Always validate user input and do not use it directly to build SQL statements

**References:**

1. https://owasp.org/www-community/attacks/SQL\_Injection

**CIS Controls:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Controls Version** | **Control** | **IG 1** | **IG 2** | **IG 3** |
| v7 | 18.2 Ensure Explicit Error Checking is Performed for All Inhouse Developed Software  For in-house developed software, ensure that explicit error checking is performed and documented for all input, including for size, data type, and acceptable ranges or formats. |  | ● | ● |
| v6 | 18.3 Sanitize Input For In-house Software  For in-house developed software, ensure that explicit error checking is performed and documented for all input, including for size, data type, and acceptable ranges or formats. |  |  |  |

#### 6.2 Ensure 'CLR Assembly Permission Set' is set to 'SAFE\_ACCESS' for All

#### CLR Assemblies (Automated)

**Profile Applicability:**

* Level 1 - Database Engine
* Level 1 - AWS RDS

**Description:**

Setting CLR Assembly Permission Sets to SAFE\_ACCESS will prevent assemblies from accessing external system resources such as files, the network, environment variables, or the registry.

**Rationale:**

Assemblies with EXTERNAL\_ACCESS or UNSAFE permission sets can be used to access sensitive areas of the operating system, steal and/or transmit data and alter the state and other protection measures of the underlying Windows Operating System.

Assemblies which are Microsoft-created (is\_user\_defined = 0) are excluded from this check as they are required for overall system functionality.

**Impact:**

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

**Audit:**

Execute the following SQL statement:

USE <database\_name>;

GO

SELECT name,

permission\_set\_desc FROM sys.assemblies

WHERE is\_user\_defined = 1;

All the returned assemblies should show SAFE\_ACCESS in the permission\_set\_desc column.

**Remediation:**

USE <database\_name>;

GO

ALTER ASSEMBLY <assembly\_name> WITH PERMISSION\_SET = SAFE;

**Default Value:**

SAFE permission is set by default.

**References:**

1. https://docs.microsoft.com/en-us/sql/relational-databases/clrintegration/security/clr-integration-code-access-security
2. https://docs.microsoft.com/en-us/sql/relational-databases/system-catalogviews/sys-assemblies-transact-sql
3. https://docs.microsoft.com/en-us/sql/t-sql/statements/alter-assembly-transactsql

**CIS Controls:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Controls Version** | **Control** | **IG 1** | **IG 2** | **IG 3** |
| v7 | 5.1 Establish Secure Configurations  Maintain documented, standard security configuration standards for all authorized operating systems and software. | ● | ● | ● |
| v6 | 18 Application Software Security  Application Software Security |  |  |  |

## 7 Encryption

These recommendations pertain to encryption-related aspects of SQL Server.

#### 7.1 Ensure 'Symmetric Key encryption algorithm' is set to 'AES\_128' or higher in non-system databases (Automated)

**Profile Applicability:**

* Level 1 - Database Engine
* Level 1 - AWS RDS

**Description:**

Per the Microsoft Best Practices, only the SQL Server AES algorithm options, AES\_128, AES\_192, and AES\_256, should be used for a symmetric key encryption algorithm.

**Rationale:**

The following algorithms (as referred to by SQL Server) are considered weak or deprecated and should no longer be used in SQL Server: DES, DESX, RC2, RC4, RC4\_128.

Many organizations may accept the Triple DES algorithms (TDEA) which use keying options 1 (3 key aka 3TDEA) or keying option 2 (2 key aka 2TDEA). In SQL Server, these are referred to as TRIPLE\_DES\_3KEY and TRIPLE\_DES respectively. Additionally, the SQL Server algorithm named DESX is actually the same implementation as the TRIPLE\_DES\_3KEY option. However, using the DESX identifier as the algorithm type has been deprecated and its usage is now discouraged.

**Impact:**

Eliminates use of weak and deprecated algorithms which may put a system at higher risk of an attacker breaking the key.

Encrypted data cannot be compressed, but compressed data can be encrypted. If you use compression, you should compress data before encrypting it.

**Audit:**

Run the following code for each individual user database:

USE <database\_name>

GO

SELECT db\_name() AS Database\_Name, name AS Key\_Name

FROM sys.symmetric\_keys

WHERE algorithm\_desc NOT IN ('AES\_128','AES\_192','AES\_256')

AND db\_id() > 4; GO

For compliance, no rows should be returned.

**Remediation:**

Refer to Microsoft SQL Server Books Online ALTER SYMMETRIC KEY entry: https://docs.microsoft.com/en-us/sql/t-sql/statements/alter-symmetric-key-transact-sql

**Default Value:**

none

**References:**

1. https://docs.microsoft.com/en-us/sql/t-sql/statements/alter-symmetric-keytransact-sql
2. http://support.microsoft.com/kb/2162020

**CIS Controls:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Controls Version** | **Control** | **IG 1** | **IG 2** | **IG 3** |
| v7 | 14.4 Encrypt All Sensitive Information in Transit  Encrypt all sensitive information in transit. |  | ● | ● |
| v6 | 14.2 Encrypt All Sensitive Information Over Less-trusted  Networks  All communication of sensitive information over less-trusted networks should be encrypted. Whenever information flows over a network with a lower trust level, the information should be encrypted. |  |  |  |

#### 7.2 Ensure Asymmetric Key Size is set to 'greater than or equal to 2048' in non-system databases (Automated)

**Profile Applicability:**

* Level 1 - Database Engine
* Level 1 - AWS RDS

**Description:**

Microsoft Best Practices recommend to use at least a 2048-bit encryption algorithm for asymmetric keys.

**Rationale:**

The RSA\_2048 encryption algorithm for asymmetric keys in SQL Server is the highest bitlevel provided and therefore the most secure available choice (other choices are RSA\_512 and RSA\_1024).

**Impact:**

The higher-bit level may result in slower performance, but reduces the likelihood of an attacker breaking the key.

Encrypted data cannot be compressed, but compressed data can be encrypted. If you use compression, you should compress data before encrypting it.

**Audit:**

Run the following code for each individual user database:

USE <database\_name>

GO

SELECT db\_name() AS Database\_Name, name AS Key\_Name

FROM sys.asymmetric\_keys

WHERE key\_length < 2048

AND db\_id() > 4; GO

For compliance, no rows should be returned.

**Remediation:**

Refer to Microsoft SQL Server Books Online ALTER ASYMMETRIC KEY entry: https://docs.microsoft.com/en-us/sql/t-sql/statements/alter-asymmetric-key-transactsql

**Default Value:**

None

**References:**

1. https://docs.microsoft.com/en-us/sql/t-sql/statements/alter-asymmetric-keytransact-sql
2. http://support.microsoft.com/kb/2162020

**CIS Controls:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Controls Version** | **Control** | **IG 1** | **IG 2** | **IG 3** |
| v7 | 14.4 Encrypt All Sensitive Information in Transit  Encrypt all sensitive information in transit. |  | ● | ● |
| v6 | 14.2 Encrypt All Sensitive Information Over Less-trusted  Networks  All communication of sensitive information over less-trusted networks should be encrypted. Whenever information flows over a network with a lower trust level, the information should be encrypted. |  |  |  |

## 8 Appendix: Additional Considerations

This appendix discusses possible configuration options for which no recommendation is being given.

#### 8.1 Ensure 'SQL Server Browser Service' is configured correctly (Manual)

**Profile Applicability:**

• Level 1 - Database Engine

**Description:**

No recommendation is being given on disabling the SQL Server Browser service.

**Rationale:**

In the case of a default instance installation, the SQL Server Browser service is disabled by default. Unless there is a named instance on the same server, there is typically no reason for the SQL Server Browser service to be running. In this case it is strongly suggested that the SQL Server Browser service remain disabled.

When it comes to named instances, given that a security scan can fingerprint a SQL Server listening on any port, it's therefore of limited benefit to disable the SQL Server Browser service.

However, if all connections against the named instance are via applications and are not visible to end users, then configuring the named instance to listening on a static port, disabling the SQL Server Browser service, and configuring the apps to connect to the specified port should be the direction taken. This follows the general practice of reducing the surface area, especially for an unneeded feature.

On the other hand, if end users are directly connecting to databases on the instance, then typically having them use *ServerName\InstanceName* is best. This requires the SQL Server Browser service to be running. Disabling the SQL Server Browser service would mean the end users would have to remember port numbers for the instances. When they don't that will generate service calls to IT staff. Given the limited benefit of disabling the service, the trade-off is probably not worth it, meaning it makes more business sense to leave the SQL Server Browser service enabled.

**Audit:**

Check the SQL Browser service's status via services.msc or similar methods.

**Remediation:**

Enable or disable the service as needed for your environment.

**Default Value:**

The SQL Server Browser service is disabled if only a default instance is installed on the server. If a named instance is installed, the default value is for the SQL Server Browser service to be configured as Automatic for startup.

**References:**

1. https://docs.microsoft.com/en-us/sql/database-engine/configure-windows/sqlserver-browser-service-database-engine-and-ssas

**CIS Controls:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Controls Version** | **Control** | **IG 1** | **IG 2** | **IG 3** |
| v7 | 9.2 Ensure Only Approved Ports, Protocols and Services Are Running  Ensure that only network ports, protocols, and services listening on a system with validated business needs, are running on each system. |  | ● | ● |
| v6 | 9.1 Limit Open Ports, Protocols, and Services  Ensure that only ports, protocols, and services with validated business needs are running on each system. |  |  |  |

# Appendix: Recommendation Summary Table

|  |  |  |  |
| --- | --- | --- | --- |
| **Control** | | **Set Correctly** | |
| Yes | No |
| **1** | **Installation, Updates and Patches** | | |
| 1.1 | Ensure Latest SQL Server Cumulative and Security Updates are Installed (Manual) | o | o |
| 1.2 | Ensure Single-Function Member Servers are Used (Manual) | o | o |
| **2** | **Surface Area Reduction** | | |
| 2.1 | Ensure 'Ad Hoc Distributed Queries' Server Configuration Option is set to '0' (Automated) | o | o |
| 2.2 | Ensure 'CLR Enabled' Server Configuration Option is set to '0' (Automated) | o | o |
| 2.3 | Ensure 'Cross DB Ownership Chaining' Server Configuration Option is set to '0' (Automated) | o | o |
| 2.4 | Ensure 'Database Mail XPs' Server Configuration Option is set to '0' (Automated) | o | o |
| 2.5 | Ensure 'Ole Automation Procedures' Server Configuration Option is set to '0' (Automated) | o | o |
| 2.6 | Ensure 'Remote Access' Server Configuration Option is set to '0' (Automated) | o | o |
| 2.7 | Ensure 'Remote Admin Connections' Server Configuration Option is set to '0' (Automated) | o | o |
| 2.8 | Ensure 'Scan For Startup Procs' Server Configuration Option is set to '0' (Automated) | o | o |
| 2.9 | Ensure 'Trustworthy' Database Property is set to 'Off' (Automated) | o | o |
| 2.10 | Ensure Unnecessary SQL Server Protocols are set to 'Disabled' (Manual) | o | o |
| 2.11 | Ensure SQL Server is configured to use non-standard ports (Automated) | o | o |
| 2.12 | Ensure 'Hide Instance' option is set to 'Yes' for Production SQL Server instances (Automated) | o | o |
| 2.13 | Ensure the 'sa' Login Account is set to 'Disabled' (Automated) | o | o |
| 2.14 | Ensure the 'sa' Login Account has been renamed (Automated) | o | o |
| 2.15 | Ensure 'AUTO\_CLOSE' is set to 'OFF' on contained databases (Automated) | o | o |

|  |  |  |  |
| --- | --- | --- | --- |
| 2.16 | Ensure no login exists with the name 'sa' (Automated) | o | o |
| 2.17 | Ensure 'clr strict security' Server Configuration Option is set to '1' (Automated) | o | o |
| **3** | **Authentication and Authorization** | | |
| 3.1 | Ensure 'Server Authentication' Property is set to 'Windows Authentication Mode' (Automated) | o | o |
| 3.2 | Ensure CONNECT permissions on the 'guest' user is Revoked within all SQL Server databases excluding the master, msdb and tempdb (Automated) | o | o |
| 3.3 | Ensure 'Orphaned Users' are Dropped From SQL Server Databases (Automated) | o | o |
| 3.4 | Ensure SQL Authentication is not used in contained databases (Automated) | o | o |
| 3.5 | Ensure the SQL Server’s MSSQL Service Account is Not an Administrator (Manual) | o | o |
| 3.6 | Ensure the SQL Server’s SQLAgent Service Account is Not an Administrator (Manual) | o | o |
| 3.7 | Ensure the SQL Server’s Full-Text Service Account is Not an Administrator (Manual) | o | o |
| 3.8 | Ensure only the default permissions specified by Microsoft are granted to the public server role (Automated) | o | o |
| 3.9 | Ensure Windows BUILTIN groups are not SQL Logins (Automated) | o | o |
| 3.10 | Ensure Windows local groups are not SQL Logins (Automated) | o | o |
| 3.11 | Ensure the public role in the msdb database is not granted access to SQL Agent proxies (Automated) | o | o |
| **4** | **Password Policies** | | |
| 4.1 | Ensure 'MUST\_CHANGE' Option is set to 'ON' for All SQL Authenticated Logins (Manual) | o | o |
| 4.2 | Ensure 'CHECK\_EXPIRATION' Option is set to 'ON' for All  SQL Authenticated Logins Within the Sysadmin Role  (Automated) | o | o |
| 4.3 | Ensure 'CHECK\_POLICY' Option is set to 'ON' for All SQL Authenticated Logins (Automated) | o | o |
| **5** | **Auditing and Logging** | | |
| 5.1 | Ensure 'Maximum number of error log files' is set to greater than or equal to '12' (Automated) | o | o |
| 5.2 | Ensure 'Default Trace Enabled' Server Configuration Option is set to '1' (Automated) | o | o |
| 5.3 | Ensure 'Login Auditing' is set to 'failed logins' (Automated) | o | o |
| 5.4 | Ensure 'SQL Server Audit' is set to capture both 'failed' and 'successful logins' (Automated) | o | o |
| **6** | **Application Development** | | |
| 6.1 | Ensure Database and Application User Input is Sanitized (Manual) | o | o |
| 6.2 | Ensure 'CLR Assembly Permission Set' is set to  'SAFE\_ACCESS' for All CLR Assemblies (Automated) | o | o |
| **7** | **Encryption** |  |  |
| 7.1 | Ensure 'Symmetric Key encryption algorithm' is set to  'AES\_128' or higher in non-system databases (Automated) | o | o |
| 7.2 | Ensure Asymmetric Key Size is set to 'greater than or equal to 2048' in non-system databases (Automated) | o | o |
| **8** | **Appendix: Additional Considerations** |  |  |
| 8.1 | Ensure 'SQL Server Browser Service' is configured correctly (Manual) | o | o |

# Appendix: Change History

|  |  |  |
| --- | --- | --- |
| Date | Version | Changes for this version |
| Nov 27, 2019 | 1.0.0 | Initial Release |
| May 15, 2020 | 1.1.0 | The text in the Rationale Statement should be two sentences (Ticket 10655) |
| May 19, 2020 | 1.1.0 | The word "as" is missing from a sentence (Ticket 10664) |
| May 19, 2020 | 1.1.0 | Minor Typo - word "has" should be "have" (Ticket 10672) |
| May 19, 2020 | 1.1.0 | Several changes to Impact Statement text to improve readability (Ticket 10673) |
| May 19, 2020 | 1.1.0 | OWASP has changed the URL in the References (Ticket 10678) |
| May 21, 2020 | 1.1.0 | Starting from SQL Server 2017 onwards only CU, and Security Updates (Ticket 10631) |
| Jun 30, 2020 | 1.1.0 | Bug Fix Release |
| Sep 10, 2020 | 1.2.0 | MSSQL 2019 Deprecation of  "xp\_cmdshell" disabled validation  (Ticket 11171) |
| Mar 2, 2021 | 1.2.0 | Error with control 7.2 Ensure  Asymmetric Key Size (Ticket 12352) |
| Mar 2, 2021 | 1.2.0 | normal database user can still view the renamed "sa" account (Ticket 10660) |
| Apr 8, 2021 | 1.2.0 | The query used for CIS-CAT does not work when dynamic ports are configured. (Ticket 11918) |
| Apr 12, 2021 | 1.2.0 | New Recommendation: Ensure 'CLR  Strict Security' Server Configuration  Option is set to '0' (Ticket 10840) |
| Apr 12, 2021 | 1.2.0 | Error in Impact Statement due to SQL Server changes - no SAFE assembly option (Ticket 10656) |
| Apr 12, 2021 | 1.2.0 | CIS SQL L1 Check 6.2 not checking all database instances (Ticket 12681) |