CIS Benchmarks

CIS Microsoft SQL Server 2017 Benchmark

v1.1.0 – 06-30-2020

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
Overview 5
Intended Audience 5
Consensus Guidance 5
Typographical Conventions 6
Assessment Status 6
Profile Definitions 7
Acknowledgements 8
Recommendations 9
1 Installation, Updates and Patches 9
1.1 Ensure Latest SQL Server Cumulative and Security Updates are Installed (Automated) 9
1.2 Ensure Single-Function Member Servers are Used (Manual) 11
2 Surface Area Reduction 13
2.1 Ensure 'Ad Hoc Distributed Queries' Server Configuration Option is set to '0' (Automated) 13
2.2 Ensure 'CLR Enabled' Server Configuration Option is set to '0' (Automated) 15
2.3 Ensure 'Cross DB Ownership Chaining' Server Configuration Option is set to '0' (Automated) 17
2.4 Ensure 'Database Mail XPs' Server Configuration Option is set to '0' (Automated) 19
2.5 Ensure 'Ole Automation Procedures' Server Configuration Option is set to '0' (Automated) 21
2.6 Ensure 'Remote Access' Server Configuration Option is set to '0' (Automated) 23
2.7 Ensure 'Remote Admin Connections' Server Configuration Option is set to '0' (Automated) 25
2.8 Ensure 'Scan For Startup Procs' Server Configuration Option is set to '0' (Automated) 27
2.9 Ensure 'Trustworthy' Database Property is set to 'Off' (Automated) 29
2.10 Ensure Unnecessary SQL Server Protocols are set to 'Disabled' (Manual) 31

2.11 Ensure SQL Server is configured to use non-standard ports (Automated) 33
2.12 Ensure 'Hide Instance' option is set to 'Yes' for Production SQL Server instances (Automated) 35
2.13 Ensure the 'sa' Login Account is set to 'Disabled' (Automated) 38
2.14 Ensure the 'sa' Login Account has been renamed (Automated) 40
2.15 Ensure 'xp_cmdshell' Server Configuration Option is set to '0' (Automated) 42
2.16 Ensure 'AUTO_CLOSE' is set to 'OFF' on contained databases (Automated) . 44
2.17 Ensure no login exists with the name 'sa' (Automated) 46
3 Authentication and Authorization 48
3.1 Ensure 'Server Authentication' Property is set to 'Windows Authentication Mode' (Automated) 48
3.2 Ensure CONNECT permissions on the 'guest' user is Revoked within all SQL Server databases excluding the master, msdb and tempdb (Automated) 50
3.3 Ensure 'Orphaned Users' are Dropped From SQL Server Databases (Automated) 52
3.4 Ensure SQL Authentication is not used in contained databases (Automated) 54
3.5 Ensure the SQL Server's MSSQL Service Account is Not an Administrator (Manual) 56
3.6 Ensure the SQL Server's SQLAgent Service Account is Not an Administrator (Manual) 58
3.7 Ensure the SQL Server's Full-Text Service Account is Not an Administrator (Manual) 60
3.8 Ensure only the default permissions specified by Microsoft are granted to the public server role (Automated) 62
3.9 Ensure Windows BUILTIN groups are not SQL Logins (Automated) 65
3.10 Ensure Windows local groups are not SQL Logins (Automated) 67
3.11 Ensure the public role in the msdb database is not granted access to SQL Agent proxies (Automated) 69
4 Password Policies 71
4.1 Ensure 'MUST_CHANGE' Option is set to 'ON' for All SQL Authenticated Logins (Manual) 71
4.2 Ensure 'CHECK_EXPIRATION' Option is set to 'ON' for All SQL Authenticated Logins Within the Sysadmin Role (Automated) 73

4.3 Ensure 'CHECK_POLICY' Option is set to 'ON' for All SQL Authenticated Logins (Automated) 75
5 Auditing and Logging 77
5.1 Ensure 'Maximum number of error log files' is set to greater than or equal to '12' (Automated) 77
5.2 Ensure 'Default Trace Enabled' Server Configuration Option is set to '1' (Automated) 80
5.3 Ensure 'Login Auditing' is set to 'failed logins' (Automated) 82
5.4 Ensure 'SQL Server Audit' is set to capture both 'failed' and 'successful logins' (Automated) 85
6 Application Development 88
6.1 Ensure Database and Application User Input is Sanitized (Manual) 88
6.2 Ensure 'CLR Assembly Permission Set' is set to 'SAFE_ACCESS' for All CLR Assemblies (Automated) 90
7 Encryption 92
7.1 Ensure 'Symmetric Key encryption algorithm' is set to 'AES_128' or higher in non-system databases (Automated) 92
7.2 Ensure Asymmetric Key Size is set to 'greater than or equal to 2048' in non-system databases (Automated) 94
8 Appendix: Additional Considerations 96
8.1 Ensure 'SQL Server Browser Service' is configured correctly (Manual) 96
Appendix: Summary Table 98
Appendix: Change History 101

Overview

This document provides prescriptive guidance for establishing a secure configuration posture for Microsoft SQL Server 2017. This guide was tested against Microsoft SQL Server 2017. 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 2017 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 intend to:

  - be practical and prudent;
  - provide a clear security benefit; and
  - not inhibit the utility of the technology beyond acceptable means.

- **Level 1 - AWS RDS**

  Items in this profile are applicable to AWS RDS and intend to:

  - be practical and prudent;
  - provide a clear security benefit; and
  - not inhibit the utility of the technology beyond acceptable means.

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
Emad Al-Mousa

Editor
Nancy Hidy Wilson
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 (Automated)

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-updates-for-microsoft-sql-server?view=sql-server-2017

Default Value:

Cumulative and security updates are not installed by default.

References:

1. https://docs.microsoft.com/en-us/sql/database-engine/install-windows/latest-updates-for-microsoft-sql-server?view=sql-server-2017
2. https://support.microsoft.com/en-us/help/4041553/sql-server-service-packs-are-discontinued-starting-from-sql-server

CIS Controls:

Version 6

4 Continuous Vulnerability Assessment and Remediation
Continuous Vulnerability Assessment and Remediation

Version 7

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.

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.

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.

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.

CIS Controls:

Version 6

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.

Version 7

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.

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/ad-hoc-distributed-queries-server-configuration-option

CIS Controls:

Version 6

9.1 Limit Open Ports, Protocols, and Services
Ensure that only ports, protocols, and services with validated business needs are running on each system.

Version 7

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.

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.

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 enabled';
```

Both value columns must show `0` to be compliant.

Remediation:

Run the following T-SQL command:

```
EXECUTE sp_configure 'clr enabled', 0;
RECONFIGURE;
```

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
```

Default Value:

By default, this option is disabled (`0`).

References:

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

CIS Controls:

Version 6

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.

Version 7

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.

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/cross-db-ownership-chaining-server-configuration-option

CIS Controls:

Version 6

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.

Version 7

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.

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/database-mail/database-mail

CIS Controls:

Version 6

18 Application Software Security
Application Software Security

Version 7

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.

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/ole-automation-procedures-server-configuration-option

CIS Controls:

Version 6

18 Application Software Security
Application Software Security

Version 7

5.1 Establish Secure Configurations
Maintain documented, standard security configuration standards for all authorized operating systems and software.

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.

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.

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.

Default Value:

By default, this option is enabled (`1`).

References:

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

CIS Controls:

Version 6

9.1 Limit Open Ports, Protocols, and Services
Ensure that only ports, protocols, and services with validated business needs are running on each system.

Version 7

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.

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/configure-windows/remote-admin-connections-server-configuration-option

Notes:

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:

Version 6

9.1 Limit Open Ports, Protocols, and Services
Ensure that only ports, protocols, and services with validated business needs are running on each system.

Version 7

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.

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.

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.

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.

Default Value:

By default, this option is disabled (`0`).

References:

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

CIS Controls:

Version 6

18 Application Software Security
Application Software Security

Version 7

5.1 Establish Secure Configurations
Maintain documented, standard security configuration standards for all authorized operating systems and software.

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/trustworthy-database-property
2. https://support.microsoft.com/it-it/help/2183687/guidelines-for-using-the-trustworthy-database-setting-in-sql-server

CIS Controls:

Version 6

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.

Version 7

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.

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.

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.

Impact:

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

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/configure-windows/enable-or-disable-a-server-network-protocol

CIS Controls:

Version 6

9.1 Limit Open Ports, Protocols, and Services

Ensure that only ports, protocols, and services with validated business needs are running on each system.

Version 7

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.

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.

Audit:

Run the following T-SQL script:

```
DECLARE @value nvarchar(256);
EXECUTE master.dbo.xp_instance_regread
        N'HKEY_LOCAL_MACHINE',
        N'SOFTWARE\Microsoft\Microsoft SQL
Server\MSSQLServer\SuperSocketNetLib\Tcp\IPAll',
        N'TcpPort',
        @value OUTPUT,
        N'no_output';

SELECT @value AS TCP_Port WHERE @value = '1433';
```

This should return no rows.

Remediation:

1. In **SQL Server Configuration Manager**, in the console pane, expand **SQL Server Network Configuration**, expand Protocols for *<InstanceName>*, and then double-click the TCP/IP protocol
2. 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.

3. 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**.
4. In the console pane, click **SQL Server Services**.
5. In the details pane, right-click **SQL Server (*<InstanceName>*)** and then click **Restart**, to stop and restart SQL Server.

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.

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/configure-windows/configure-a-server-to-listen-on-a-specific-tcp-port

Notes:

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

CIS Controls:

Version 6

9 Limitation and Control of Network Ports, Protocols, and Services
Limitation and Control of Network Ports, Protocols, and Services

Version 7

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.

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.

Audit:

Perform either the GUI or T-SQL method shown:

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.

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:

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.

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;
```

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.

Default Value:

By default, SQL Server instances are not hidden.

References:

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

CIS Controls:

Version 6

9 Limitation and Control of Network Ports, Protocols, and Services
Limitation and Control of Network Ports, Protocols, and Services

Version 7

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.

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.

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
```

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.

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-catalog-views/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-an-authentication-mode

Notes:

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

CIS Controls:

Version 6

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.

Version 7

16.8 Disable Any Unassociated Accounts

Disable any account that cannot be associated with a business process or business owner.

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.

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>;
```

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.

Default Value:

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

References:

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

Notes:

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

CIS Controls:

Version 6

5 Controlled Use of Administration Privileges
Controlled Use of Administration Privileges

Version 7

5.1 Establish Secure Configurations
Maintain documented, standard security configuration standards for all authorized operating systems and software.

2.15 Ensure 'xp_cmdshell' Server Configuration Option is set to '0' (Automated)

Profile Applicability:

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

Description:

The `xp_cmdshell` option controls whether the `xp_cmdshell` extended stored procedure can be used by an authenticated SQL Server user to execute operating-system command shell commands and return results as rows within the SQL client.

Rationale:

The `xp_cmdshell` procedure is commonly used by attackers to read or write data to/from the underlying Operating System of a database 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 = 'xp_cmdshell';
```

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 'xp_cmdshell', 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/system-stored-procedures/xp-cmdshell-transact-sql
2. https://docs.microsoft.com/en-us/sql/database-engine/configure-windows/xp-cmdshell-server-configuration-option

CIS Controls:

Version 6

18 Application Software Security
Application Software Security

Version 7

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.

2.16 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/security-best-practices-with-contained-databases

CIS Controls:

Version 6

18 Application Software Security
Application Software Security

Version 7

5.1 Establish Secure Configurations
Maintain documented, standard security configuration standards for all authorized operating systems and software.

2.17 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.

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
```

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.

Default Value:

The login with `principal_id = 1` is named `sa` by default.

CIS Controls:

Version 6

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.

Version 7

5.1 Establish Secure Configurations
Maintain documented, standard security configuration standards for all authorized operating systems and software.

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.

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:

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**.

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.

Impact:

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

Default Value:

Windows Authentication Mode

References:

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

CIS Controls:

Version 6

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.

Version 7

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.

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.

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;
```

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.

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-based-management/guest-permissions-on-user-databases

Notes:

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:

Version 6

16 Account Monitoring and Control
Account Monitoring and Control

Version 7

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.

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/troubleshoot-orphaned-users-sql-server

CIS Controls:

Version 6

16 Account Monitoring and Control
Account Monitoring and Control

Version 7

16.8 Disable Any Unassociated Accounts
Disable any account that cannot be associated with a business process or business owner.

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.

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.

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.

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/security-best-practices-with-contained-databases

CIS Controls:

Version 6

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).

Version 7

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.

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 `MSSQL$<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.

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.

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.

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/configure-windows/configure-windows-service-accounts-and-permissions

CIS Controls:

Version 6

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.

Version 7

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.

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.

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.

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.

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/configure-windows/configure-windows-service-accounts-and-permissions

CIS Controls:

Version 6

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.

Version 7

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.

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.

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.

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.

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/configure-windows/configure-windows-service-accounts-and-permissions

CIS Controls:

Version 6

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.

Version 7

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.

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.

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
```

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.

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/relational-databases/security/authentication-access/server-level-roles
2. https://docs.microsoft.com/en-us/sql/relational-databases/security/authentication-access/server-level-roles#permissions-of-fixed-server-roles

CIS Controls:

Version 6

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.

Version 7

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.

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.

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
```

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.

Default Value:

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

CIS Controls:

Version 6

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.

Version 7

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.

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.

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
```

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.

Default Value:

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

CIS Controls:

Version 6

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.

Version 7

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.

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.

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
```

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.

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-configuring-sql-server-agent-proxy-account

CIS Controls:

Version 6

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.

Version 7

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.

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.

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;
```

Impact:

`CHECK_EXPIRATION` and `CHECK_POLICY` options must both be `ON`. End users must have the means (application) to change the password when forced.

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:

Version 6

16 Account Monitoring and Control
Account Monitoring and Control

Version 7

4.2 Change Default Passwords
Before deploying any new asset, change all default passwords to have values consistent with administrative level accounts.

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.

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;
```

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.

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/password-policy?view=sql-server-2017

CIS Controls:

Version 6

16.2 All Accounts Have A Monitored Expiration Date
Ensure that all accounts have an expiration date that is monitored and enforced.

Version 7

16.10 Ensure All Accounts Have An 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.

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.

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.

Default Value:

`CHECK_POLICY` is `ON`

References:

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

CIS Controls:

Version 6

16 Account Monitoring and Control
Account Monitoring and Control

Version 7

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.

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.

Audit:

Perform either the GUI or T-SQL method shown:

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`

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:

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`

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>;
```

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.

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/scm-services-configure-sql-server-error-logs

CIS Controls:

Version 6

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.

Version 7

6.4 Ensure adequate storage for logs

Ensure that all systems that store logs have adequate storage space for the logs generated.

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/configure-windows/default-trace-enabled-server-configuration-option

CIS Controls:

Version 6

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.

Version 7

6.2 Activate audit logging

Ensure that local logging has been enabled on all systems and networking devices.

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.

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.

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:

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.

T-SQL Method

1. Run:

```
EXEC xp_instance_regwrite N'HKEY_LOCAL_MACHINE',
N'Software\Microsoft\MSSQLServer\MSSQLServer', N'AuditLevel',
REG_DWORD, 2
```

2. Restart the SQL Server instance.

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.

Default Value:

By default, only failed login attempts are captured.

References:

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

CIS Controls:

Version 6

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 flagging the use of the user's credentials from a computer other than computers on which the user generally works.

Version 7

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.

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.

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:

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**.

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.

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.

Default Value:

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

References:

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

CIS Controls:

Version 6

5.5 Log Failed Administrative Login Attempts
Configure systems to issue a log entry and alert on any unsuccessful login to an administrative account.

Version 7

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.

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.

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

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.

References:

1. https://owasp.org/www-community/attacks/SQL_Injection

CIS Controls:

Version 6

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.

Version 7

18.2 Ensure Explicit Error Checking is Performed for All In-house 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.

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.

Audit:

Execute the following SQL statement:

```
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:

```
ALTER ASSEMBLY <assembly_name> WITH PERMISSION_SET = SAFE;
```

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.

Default Value:

`SAFE` permission is set by default.

References:

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

CIS Controls:

Version 6

18 Application Software Security
Application Software Security

Version 7

5.1 Establish Secure Configurations
Maintain documented, standard security configuration standards for all authorized operating systems and software.

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.

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

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.

Default Value:

none

References:

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

CIS Controls:

Version 6

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.

Version 7

14.4 Encrypt All Sensitive Information in Transit
Encrypt all sensitive information in transit.

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 bit-level provided and therefore the most secure available choice (other choices are `RSA_512` and `RSA_1024`).

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-transact-sql

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.

Default Value:

None

References:

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

CIS Controls:

Version 6

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.

Version 7

14.4 Encrypt All Sensitive Information in Transit
Encrypt all sensitive information in transit.

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/sql-server-browser-service-database-engine-and-ssas

CIS Controls:

Version 6

9.1 Limit Open Ports, Protocols, and Services
Ensure that only ports, protocols, and services with validated business needs are running on each system.

Version 7

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.

Appendix: Summary Table

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

3.1	Ensure 'Server Authentication' Property is set to 'Windows Authentication Mode' (Automated)	☐	☐
3.2	Ensure CONNECT permissions on the 'guest' user is Revoked within all SQL Server databases excluding the master, msdb and tempdb (Automated)	☐	☐
3.3	Ensure 'Orphaned Users' are Dropped From SQL Server Databases (Automated)	☐	☐
3.4	Ensure SQL Authentication is not used in contained databases (Automated)	☐	☐
3.5	Ensure the SQL Server's MSSQL Service Account is Not an Administrator (Manual)	☐	☐
3.6	Ensure the SQL Server's SQLAgent Service Account is Not an Administrator (Manual)	☐	☐
3.7	Ensure the SQL Server's Full-Text Service Account is Not an Administrator (Manual)	☐	☐
3.8	Ensure only the default permissions specified by Microsoft are granted to the public server role (Automated)	☐	☐
3.9	Ensure Windows BUILTIN groups are not SQL Logins (Automated)	☐	☐
3.10	Ensure Windows local groups are not SQL Logins (Automated)	☐	☐
3.11	Ensure the public role in the msdb database is not granted access to SQL Agent proxies (Automated)	☐	☐
4	**Password Policies**		
4.1	Ensure 'MUST_CHANGE' Option is set to 'ON' for All SQL Authenticated Logins (Manual)	☐	☐
4.2	Ensure 'CHECK_EXPIRATION' Option is set to 'ON' for All SQL Authenticated Logins Within the Sysadmin Role (Automated)	☐	☐
4.3	Ensure 'CHECK_POLICY' Option is set to 'ON' for All SQL Authenticated Logins (Automated)	☐	☐
5	**Auditing and Logging**		
5.1	Ensure 'Maximum number of error log files' is set to greater than or equal to '12' (Automated)	☐	☐
5.2	Ensure 'Default Trace Enabled' Server Configuration Option is set to '1' (Automated)	☐	☐
5.3	Ensure 'Login Auditing' is set to 'failed logins' (Automated)	☐	☐
5.4	Ensure 'SQL Server Audit' is set to capture both 'failed' and 'successful logins' (Automated)	☐	☐
6	**Application Development**		
6.1	Ensure Database and Application User Input is Sanitized (Manual)	☐	☐
6.2	Ensure 'CLR Assembly Permission Set' is set to 'SAFE_ACCESS' for All CLR Assemblies (Automated)	☐	☐
7	**Encryption**		

7.1	Ensure 'Symmetric Key encryption algorithm' is set to 'AES_128' or higher in non-system databases (Automated)	☐	☐
7.2	Ensure Asymmetric Key Size is set to 'greater than or equal to 2048' in non-system databases (Automated)	☐	☐
8	**Appendix: Additional Considerations**		
8.1	Ensure 'SQL Server Browser Service' is configured correctly (Manual)	☐	☐

Appendix: Change History

Date	Version	Changes for this version
May 31, 2019	1.0.0	Initial Release
May 19, 2020	1.1.0	The word "as" is missing from a sentence (Ticket 10821)
May 19, 2020	1.1.0	Minor Typo - word "has" should be "have" (Ticket 10813)
May 19, 2020	1.1.0	Several changes to Impact Statement text to improve readability (Ticket 10830)
May 19, 2020	1.1.0	The text in the Rationale Statement should be two sentences (Ticket 10818)
May 19, 2020	1.1.0	OWASP has changed the URL in the References (Ticket 10744)
May 22, 2020	1.1.0	Starting from SQL Server 2017 onwards only CU, and Security Updates (Ticket 10819)
Jun 30, 2020	1.1.0	Bugfix Release