CENTER FOR
INTERNET SECURITY

CIS Microsoft SQL Server 2014 Benchmark

v1.1.0 - 09-24-2015

The CIS Security Benchmarks division provides consensus-oriented information security products, services, tools, metrics, suggestions, and recommendations (the "SB Products") as a public service to Internet users worldwide. Downloading or using SB Products in any way signifies and confirms your acceptance of and your binding agreement to these CIS Security Benchmarks Terms of Use.

CIS SECURITY BENCHMARKS TERMS OF USE

BOTH CIS SECURITY BENCHMARKS DIVISION MEMBERS AND NON-MEMBERS MAY:

- Download, install, and use each of the SB Products on a single computer, and/or
- Print one or more copies of any SB Product that is in a .txt, .pdf, .doc, .mcw, or .rtf format, but only if each such copy is printed in its entirety and is kept intact, including without limitation the text of these CIS Security Benchmarks Terms of Use.

UNDER THE FOLLOWING TERMS AND CONDITIONS:

- **SB Products Provided As Is**. CIS is providing the SB Products "as is" and "as available" without: (1) any representations, warranties, or covenants of any kind whatsoever (including the absence of any warranty regarding: (a) the effect or lack of effect of any SB Product on the operation or the security of any network, system, software, hardware, or any component of any of them, and (b) the accuracy, utility, reliability, timeliness, or completeness of any SB Product); or (2) the responsibility to make or notify you of any corrections, updates, upgrades, or fixes.
- **Intellectual Property and Rights Reserved**. You are not acquiring any title or ownership rights in or to any SB Product, and full title and all ownership rights to the SB Products remain the exclusive property of CIS. All rights to the SB Products not expressly granted in these Terms of Use are hereby reserved.
- **Restrictions**. You acknowledge and agree that you may not: (1) decompile, dis-assemble, alter, reverse engineer, or otherwise attempt to derive the source code for any software SB Product that is not already in the form of source code; (2) distribute, redistribute, sell, rent, lease, sublicense or otherwise transfer or exploit any rights to any SB Product in any way or for any purpose; (3) post any SB Product on any website, bulletin board, ftp server, newsgroup, or other similar mechanism or device; (4) remove from or alter these CIS Security Benchmarks Terms of Use on any SB Product; (5) remove or alter any proprietary notices on any SB Product; (6) use any SB Product or any component of an SB Product with any derivative works based directly on an SB Product or any component of an SB Product; (7) use any SB Product or any component of an SB Product with other products or applications that are directly and specifically dependent on such SB Product or any component for any part of their functionality; (8) represent or claim a particular level of compliance or consistency with any SB Product; or (9) facilitate or otherwise aid other individuals or entities in violating these CIS Security Benchmarks Terms of Use.
- **Your Responsibility to Evaluate Risks**. You acknowledge and agree that: (1) no network, system, device, hardware, software, or component can be made fully secure; (2) you have the sole responsibility to evaluate the risks and benefits of the SB Products to your particular circumstances and requirements; and (3) CIS is not assuming any of the liabilities associated with your use of any or all of the SB Products.
- **CIS Liability**. You acknowledge and agree that neither CIS nor any of its employees, officers, directors, agents or other service providers has or will have any liability to you whatsoever (whether based in contract, tort, strict liability or otherwise) for any direct, indirect, incidental, consequential, or special damages that arise out of or are connected in any way with your use of any SB Product.
- **Indemnification**. You agree to indemnify, defend, and hold CIS and all of CIS's employees, officers, directors, agents and other service providers harmless from and against any liabilities, costs and expenses incurred by any of them in connection with your violation of these CIS Security Benchmarks Terms of Use.
- **Jurisdiction**. You acknowledge and agree that: (1) these CIS Security Benchmarks Terms of Use will be governed by and construed in accordance with the laws of the State of Maryland; (2) any action at law or in equity arising out of or relating to these CIS Security Benchmarks Terms of Use shall be filed only in the courts located in the State of Maryland; and (3) you hereby consent and submit to the personal jurisdiction of such courts for the purposes of litigating any such action.
- **U.S. Export Control and Sanctions laws**. Regarding your use of the SB Products with any non-U.S. entity or country, you acknowledge that it is your responsibility to understand and abide by all U.S. sanctions and export control laws as set from time to time by the U.S. Bureau of Industry and Security (BIS) and the U.S. Office of Foreign Assets Control (OFAC).

SPECIAL RULES FOR CIS MEMBER ORGANIZATIONS: CIS reserves the right to create special rules for: (1) CIS Members; and (2) Non-Member organizations and individuals with which CIS has a written contractual relationship. CIS hereby grants to each CIS Member Organization in good standing the right to distribute the SB Products within such Member's own organization, whether by manual or electronic means. Each such Member Organization acknowledges and agrees that the foregoing grants in this paragraph are subject to the terms of such Member's membership arrangement with CIS and may, therefore, be modified or terminated by CIS at any time.

Table of Contents

Overview

This document provides prescriptive guidance for establishing a secure configuration posture for Microsoft SQL Server 2014. This guide was tested against Microsoft SQL Server 2014. 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 2014 on a Microsoft Windows platform.

Consensus Guidance

This benchmark was created using a consensus review process comprised 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://community.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

Scoring Information

A scoring status indicates whether compliance with the given recommendation impacts the assessed target's benchmark score. The following scoring statuses are used in this benchmark:

Scored

Failure to comply with "Scored" recommendations will decrease the final benchmark score. Compliance with "Scored" recommendations will increase the final benchmark score.

Not Scored

Failure to comply with "Not Scored" recommendations will not decrease the final benchmark score. Compliance with "Not Scored" recommendations will not increase the final benchmark score.

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.

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
Blake Frantz

Editor
Nancy Hidy Wilson
Brian Kelley MCSE, CISA, Security+, Microsoft MVP - SQL Server

CIS also thanks the following contributors to the CIS Microsoft SQL Server 2012 Benchmark, on which this benchmark is heavily based: Kevvie Fowler, Tran Thanh Chien, Masoud Sultan, and Dorothy Feistner.

Recommendations

1 Installation, Updates and Patches

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

1.1 Install the Latest SQL Server Service Packs and Hotfixes (Scored)

Profile Applicability:

- Level 1 - Database Engine

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 hotfix which is a single patch, a cumulative update which is a small group of patches or a service pack which is a large collection 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 service pack level, run the following code snippet.

```
SELECT SERVERPROPERTY('ProductLevel') as SP_installed,
SERVERPROPERTY('ProductVersion') as Version;
```

First column returns the installed Service Pack level, the second is the exact build number.

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:

- Hotfixes and Cumulative updates: http://blogs.msdn.com/b/sqlreleaseservices/
- Service Packs: https://support.microsoft.com/en-us/kb/2958069

Default Value:

Service packs and patches are not installed by default.

References:

1. https://support.microsoft.com/en-us/kb/2958069

1.2 Install on dedicated single-function member servers (Not Scored)

Profile Applicability:

- Level 1 - Database Engine

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.

2 Surface Area Reduction

SQL Server offers various configuration options, some of them can be controlled by the sp_configure stored procedures. This section contains the listing of the corresponding recommendations.

2.1 Set the 'Ad Hoc Distributed Queries' Server Configuration Option to 0 (Scored)

Profile Applicability:

- Level 1 - Database Engine

Description:

Ad Hoc Distributed Queries Allow 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://msdn.microsoft.com/en-us/library/ms187569(v=sql.120).aspx

2.2 Set the 'CLR Enabled' Server Configuration Option to 0 (Scored)

Profile Applicability:

- Level 1 - Database Engine

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.

Remediation:

Run the following T-SQL command:

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

Default Value:

By default, this option is disabled.

2.3 Set the 'Cross DB Ownership Chaining' Server Configuration Option to 0 (Scored)

Profile Applicability:

- Level 1 - Database Engine

Description:

This option allows controlling 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 <dbname> 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.

Remediation:

Run the following T-SQL command:

```
EXECUTE sp_configure 'Cross db ownership chaining', 0;
RECONFIGURE;
GO
```

Default Value:

0 (disabled)

References:

1. http://msdn.microsoft.com/en-us/library/ms188694(v=sql.120).aspx

2.4 Set the 'Database Mail XPs' Server Configuration Option to 0 (Scored)

Profile Applicability:

- Level 1 - Database Engine

Description:

This option controls the generation and transmission of email messages from SQL Server.

Rationale:

Disabling Database Mail 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.

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.

2.5 Set the 'Ole Automation Procedures' Server Configuration Option to 0 (Scored)

Profile Applicability:

- Level 1 - Database Engine

Description:

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:

```
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';
```

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:

0 (disabled)

References:

1. http://msdn.microsoft.com/en-us/library/ms191188(v=sql.120).aspx

2.6 Set the 'Remote Access' Server Configuration Option to 0 (Scored)

Profile Applicability:

- Level 1 - Database Engine

Description:

Enables 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;
```

Default Value:

1 (enabled)

References:

1. http://msdn.microsoft.com/en-us/library/ms187660(v=sql.120).aspx

2.7 Set the 'Remote Admin Connections' Server Configuration Option to 0 (Scored)

Profile Applicability:

- Level 1 - Database Engine

Description:

This setting 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.

Remediation:

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

```
EXECUTE sp_configure 'Remote admin connections', 0;
RECONFIGURE;
GO
```

Default Value:

0 (disabled)

References:

1. http://msdn.microsoft.com/en-us/library/ms190468(v=sql.120).aspx

2.8 Set the 'Scan For Startup Procs' Server Configuration Option to 0 (Scored)

Profile Applicability:

- Level 1 - Database Engine

Description:

This option 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;
```

Impact:

Setting Scan for Startup Procedures to 0 will prevent certain audit traces and other commonly used monitoring SPs 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:

0 (disabled)

References:

1. http://msdn.microsoft.com/en-us/library/ms179460(v=sql.120).aspx

2.9 Set the 'Trustworthy' Database Property to Off (Scored)

Profile Applicability:

- Level 1 - Database Engine

Description:

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

Rationale:

Provides protection from malicious CLR assemblies or extended procedures.

Audit:

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

```
SELECT name
FROM sys.databases
WHERE is_trustworthy_on = 1
AND name != 'msdb'
AND state = 0;
```

Remediation:

Execute the following statement against the database:

```
ALTER DATABASE <dbname>
SET TRUSTWORTHY OFF;
```

Default Value:

OFF

References:

1. http://msdn.microsoft.com/en-us/library/ms187861(v=sql.120).aspx

2.10 Disable Unnecessary SQL Server Protocols (Not Scored)

Profile Applicability:

- Level 1 - Database Engine

Description:

SQL Server supports Shared Memory, Named Pipes, TCP/IP and VIA 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 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 commerical SQL Server 2008 instances.

References:

1. http://msdn.microsoft.com/en-us/library/ms191294(v=sql.120).aspx

2.11 Configure SQL Server to use non-standard ports (Not Scored)

Profile Applicability:

- Level 1 - Database Engine

Description:

If enabled, the default SQL Server instance will be assigned a default port of TCP:1433 for TCP/IP communication. Administrators can also 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.

Rationale:

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

Audit:

Open a powershell window and run the following command:

```
PS C:\>netstat -ano|select-string 1433.+listening
```

This should return no lines. If any lines returned, check the process id in the last column if it's a SQL Server instance.

Remediation:

1. In SQL Server Configuration Manager, in the console pane, expand SQL Server Network Configuration, expand Protocols for , and then double-click the TCP/IP or VIA 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. Change the TCP Port field from 1433 to another 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 () and then click Restart, to stop and restart SQL Server.

Impact:

Changing the default port will force DAC (Default Administrator Connection) to listen on a random port. Also, it might make benign applications, such as application firewalls, require special configuration.

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. http://msdn.microsoft.com/en-us/library/ms177440(v=sql.120).aspx

2.12 Set the 'Hide Instance' option to 'Yes' for Production SQL Server instances (Scored)

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:

1. In SQL Server Configuration Manager, expand SQL Server Network Configuration, right-click Protocols for <server instance>, 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.

Remediation:

1. In SQL Server Configuration Manager, expand SQL Server Network Configuration, right-click Protocols for <server instance>, 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.

Default Value:

By default, SQL Server instances are not hidden.

References:

1. http://msdn.microsoft.com/en-us/library/ms179327(v=sql.120).aspx

2.13 Disable the 'sa' Login Account (Scored)

Profile Applicability:

- Level 1 - Database Engine

Description:

The `sa` account is a widely known and often widely used SQL Server account with sysadmin privileges.

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.

```
SELECT name, is_disabled
FROM sys.server_principals
WHERE sid = 0x01;
```

An is_disabled value of 1 indicates the account is currently disabled.

Remediation:

Execute the following query:

```
ALTER LOGIN sa DISABLE;
```

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 for authenticating to the database server and executing required tasks or functions.

Default Value:

By default the 'sa' login account is enabled.

References:

1. http://msdn.microsoft.com/en-us/library/ms188786(v=sql.120).aspx
2. http://msdn.microsoft.com/en-us/library/ms189828(v=sql.120).aspx

2.14 Rename the 'sa' Login Account (Scored)

Profile Applicability:

- Level 1 - Database Engine

Description:

The sa account is a widely known and often widely used SQL Server account with sysadmin privileges.

Rationale:

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

Audit:

Use the following syntax to determine if the sa account is renamed.

```
SELECT name
FROM sys.server_principals
WHERE sid = 0x01;
```

A name of sa indicates the account has not been renamed.

Remediation:

Replace the different_user value within the below syntax and execute 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 account. However, if this has been done renaming the sa account will prevent scripts and applications for authenticating to the database server and executing required tasks or functions.

Default Value:

By default the '`sa`'account name is 'sa'

References:

1. http://msdn.microsoft.com/en-us/library/ms144284.aspx

2.15 Set the 'xp_cmdshell' Server Configuration Option to 0 (Scored)

Profile Applicability:

- Level 1 - Database Engine

Description:

The xp_cmdshell procedure allows an authenticated SQL Server user to execute operating-system command shell commands and return results as rows within the SQL client.

Rationale:

xp_cmdshell is commonly used by attackers to read or write data to/from the underlying Operating System of a database server.

Audit:

Run the following code snippet to determine if the xp_cmdshell system stored procedure is enabled:

```
EXECUTE sp_configure 'show advanced options',1;
RECONFIGURE WITH OVERRIDE;
EXECUTE sp_configure 'xp_cmdshell';
```

A run value of 0 indicates that the xp_cmdshell option is disabled. If the option is enabled, run the following code snippet to disable this option:

```
EXECUTE sp_configure 'show advanced options',1;
RECONFIGURE WITH OVERRIDE;
EXECUTE sp_configure 'xp_cmdshell',0;
RECONFIGURE WITH OVERRIDE;
```

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:

0 (disabled)

References:

1. http://msdn.microsoft.com/en-us/library/ms175046(v=sql.120).aspx
2. http://msdn.microsoft.com/en-us/library/ms190693(v=sql.120).aspx

2.16 Set AUTO_CLOSE OFF on contained databases (Scored)

Profile Applicability:

- Level 1 - Database Engine

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:

Opening contained databases to authenticate a user 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;
```

Remediation:

Perform the following to establish the prescribed state for a given contained database:

```
ALTER DATABASE <database_name> SET AUTO_CLOSE OFF;
```

Default Value:

AUTO_CLOSE is off (Is_auto_close_on = 0)

References:

1. http://msdn.microsoft.com/en-us/library/ff929055.aspx

2.17 Verify No Login Has the Name 'sa' (Scored)

Profile Applicability:

- Level 1 - Database Engine

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' when the original 'sa' login (sid 0x01) has been renamed.

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 there is an account named sa.

```
SELECT sid, name,

FROM sys.server_principals

WHERE L.name = 'sa'

AND L.sid <> 0x01;
```

This query should not return any rows.

Remediation:

Execute the following query:

```
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 a sid of 0x01 is named 'sa' by default.

3 Authentication and Authorization

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

3.1 Set The 'Server Authentication' Property To Windows Authentication mode (Scored)

Profile Applicability:

- Level 1 - Database Engine

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:

```
xp_loginconfig 'login mode';
```

A config_value of Windows NT Authentication indicates the Server Authentication property is set to Windows Authentication mode

Remediation:

Perform the following steps:

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.

Default Value:

Windows Authentication Mode

References:

1. http://msdn.microsoft.com/en-us/library/ms188470(v=sql.120).aspx

3.2 Revoke CONNECT permissions on the 'guest user' within all SQL Server databases excluding the master, msdb and tempdb (Scored)

Profile Applicability:

- Level 1 - Database Engine

Description:

Removes the right of guest users to connect to SQL Server user databases.

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 in each database in the instance to determine if the guest user has CONNECT permission.

```
USE [database_name];
GO
SELECT DB_NAME() AS DBName, dpr.name, dpe.permission_name
FROM sys.database_permissions dpe
JOIN sys.database_principals dpr
```

```
ON dpe.grantee_principal_id=dpr.principal_id
WHERE dpr.name='guest'
AND dpe.permission_name='CONNECT';
```

Remediation:

The following code snippet revokes CONNECT permissions from the guest user in a database:

```
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. http://msdn.microsoft.com/en-us/library/bb402861(v=sql.120).aspx

3.3 Drop Orphaned Users From SQL Server Databases (Scored)

Profile Applicability:

- Level 1 - Database Engine

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 to identify orphan users:

```
EXEC sp_change_users_login @Action='Report';
```

Remediation:

Run the following T-SQL query to remove an orphan user:

```
DROP USER <username>;
```

References:

1. http://msdn.microsoft.com/en-us/library/ms175475(v=sql.120).aspx

3.4 Do not use SQL Authentication in contained databases (Scored)

Profile Applicability:

- Level 1 - Database Engine

Description:

Contained databases do not enforce password complexity rules.

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 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 Authentication.

References:

1. http://msdn.microsoft.com/en-us/library/ff929055.aspx

4 Password Policies

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

4.1 Set the 'MUST_CHANGE' Option to ON for All SQL Authenticated Logins (Not Scored)

Profile Applicability:

- Level 1 - Database Engine

Description:

SQL Server will prompt for an updated password the first time the altered login is used.

Rationale:

Enforcing password change will prevent the account administrators or anyone accessing the initial password to misuse 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

Remediation:

Set the MUST_CHANGE option for SQL Authenticated logins

```
ALTER LOGIN login_name WITH PASSWORD = password_value MUST_CHANGE;
```

Impact:

CHECK_EXPIRATION and CHECK_POLICY options must both be ON

Default Value:

ON

References:

1. http://msdn.microsoft.com/en-us/library/ms189828(v=sql.120).aspx

4.2 Set the 'CHECK_EXPIRATION' Option to ON for All SQL Authenticated Logins Within the Sysadmin Role (Scored)

Profile Applicability:

- Level 1 - Database Engine

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:

```
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:

```
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.

In regards to 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 has this failing.

Also, keep in mind that the password policy is taken from the computer's local policy, which will take from the Default Domain Policy setting. Many organizations have a different password policy with regards to service accounts. These are handled in AD by setting the account's password not to expire and having some other process track when they need 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. http://msdn.microsoft.com/en-us/library/ms161959(v=sql.120).aspx

4.3 Set the 'CHECK_POLICY' Option to ON for All SQL Authenticated Logins (Scored)

Profile Applicability:

- Level 1 - Database Engine

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 an unusable. If no rows are returned then either no SQL Authenticated logins exist or they all have 'CHECK_POLICY' ON.

Remediation:

```
ALTER LOGIN [login_name] WITH CHECK_POLICY = ON;
```

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 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. http://msdn.microsoft.com/en-us/library/ms161959(v=sql.120).aspx

5 Auditing and Logging

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

5.1 Set the 'Maximum number of error log files' setting to greater than or equal to 12 (Not Scored)

Profile Applicability:

- Level 1 - Database Engine

Description:

SQL Server errorlog files must be protected from loss. The log files must be backed up before they are overwritten.

Rationale:

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

Audit:

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

Remediation:

Adjust the number of logs to prevent data loss. The default value of 6 may be insufficient for a production environment.

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

Default Value:

6 SQL Server error logs are retained by default.

References:

1. http://msdn.microsoft.com/en-us/library/ms177285(v=sql.120).aspx

5.2 Set the 'Default Trace Enabled' Server Configuration Option to 1 (Scored)

Profile Applicability:

- Level 1 - Database Engine

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. http://msdn.microsoft.com/en-us/library/ms175513(v=sql.120).aspx

5.3 Set 'Login Auditing' to failed logins (Not Scored)

Profile Applicability:

- Level 1 - Database Engine

Description:

Setting logs to record only failed login SQL Server authentication attempts. This is the default setting for SQL Server.

Rationale:

Logging failed logins provides key information that can be used to detect\confirm password guessing attacks. While logging successful login attempts can be used to confirm server access during forensic investigations, we recommend using Audit. This minimizes the writes to the ERRORLOG for SQL Server, which reduces the amount of noise for when a DBA is trying to troubleshoot an issue.

Audit:

```
XP_loginconfig 'audit level';
```

A config_value of 'failure' indicates a server login auditing setting of 'Failed logins only'.

Remediation:

Perform the following steps to set the level of auditing:

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.

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.

Default Value:

By default, only failed login attempted are captured.

References:

1. http://technet.microsoft.com/en-us/library/ms188470(v=sql.120).aspx

5.4 Use SQL Server Audit to capture both failed and successful logins (Not Scored)

Profile Applicability:

- Level 1 - Database Engine

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 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:

Via the SSMS GUI Interface:

1. Expand the SQL Server in Object Explorer.
2. Expand the Secuirty 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 under 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.

Via T-SQL:

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
```

Please 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 is no audit object tracking login events. Also, successful logins aren't normally tracked.

References:

1. https://msdn.microsoft.com/en-us/library/cc280525(v=sql.120).aspx

6 Application Development

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

6.1 Sanitize Database and Application User Input (Not Scored)

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://www.owasp.org/index.php/SQL_Injection
2. http://msdn.microsoft.com/en-us/library/ms161953(v=sql.120).aspx

6.2 Set the 'CLR Assembly Permission Set' to SAFE_ACCESS for All CLR Assemblies (Scored)

Profile Applicability:

- Level 1 - Database Engine

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. http://msdn.microsoft.com/en-us/library/ms345101(v=sql.120).aspx
2. http://msdn.microsoft.com/en-us/library/ms189790(v=sql.120).aspx
3. http://msdn.microsoft.com/en-us/library/ms186711(v=sql.120).aspx

7 Encryption

This setting contains recommendations pertaining to encryption-related aspects of MS SQL.

7.1 Ensure Symmetric Key encryption algorithm is AES_128 or higher in non-system databases (Scored)

Profile Applicability:

- Level 1 - Database Engine

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 [dbname]
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:
http://msdn.microsoft.com/en-US/library/ms189440.aspx

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

7.2 Ensure asymmetric key size is greater than or equal to 2048 in non-system databases (Scored)

Profile Applicability:

- Level 1 - Database Engine

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 [dbname]
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: http://msdn.microsoft.com/en-us/library/ms187311.aspx

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

8 Appendix: Additional Considerations

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

8.1 SQL Server Browser Service (Not Scored)

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 no typically 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:

[This section intentionally left blank.]

Remediation:

[This section intentionally left blank.]

Impact:

[This section intentionally left blank.]

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.

Control		Set Correctly	
		Yes	No
1	**Installation, Updates and Patches**		
1.1	Install the Latest SQL Server Service Packs and Hotfixes (Scored)	☐	☐
1.2	Install on dedicated single-function member servers (Not Scored)	☐	☐
2	**Surface Area Reduction**		
2.1	Set the 'Ad Hoc Distributed Queries' Server Configuration Option to 0 (Scored)	☐	☐
2.2	Set the 'CLR Enabled' Server Configuration Option to 0 (Scored)	☐	☐
2.3	Set the 'Cross DB Ownership Chaining' Server Configuration Option to 0 (Scored)	☐	☐

2.4	Set the 'Database Mail XPs' Server Configuration Option to 0 (Scored)	☐	☐
2.5	Set the 'Ole Automation Procedures' Server Configuration Option to 0 (Scored)	☐	☐
2.6	Set the 'Remote Access' Server Configuration Option to 0 (Scored)	☐	☐
2.7	Set the 'Remote Admin Connections' Server Configuration Option to 0 (Scored)	☐	☐
2.8	Set the 'Scan For Startup Procs' Server Configuration Option to 0 (Scored)	☐	☐
2.9	Set the 'Trustworthy' Database Property to Off (Scored)	☐	☐
2.10	Disable Unnecessary SQL Server Protocols (Not Scored)	☐	☐
2.11	Configure SQL Server to use non-standard ports (Not Scored)	☐	☐
2.12	Set the 'Hide Instance' option to 'Yes' for Production SQL Server instances (Scored)	☐	☐
2.13	Disable the 'sa' Login Account (Scored)	☐	☐
2.14	Rename the 'sa' Login Account (Scored)	☐	☐
2.15	Set the 'xp_cmdshell' Server Configuration Option to 0 (Scored)	☐	☐
2.16	Set AUTO_CLOSE OFF on contained databases (Scored)	☐	☐
2.17	Verify No Login Has the Name 'sa' (Scored)	☐	☐
3	**Authentication and Authorization**		
3.1	Set The 'Server Authentication' Property To Windows Authentication mode (Scored)	☐	☐
3.2	Revoke CONNECT permissions on the 'guest user' within all SQL Server databases excluding the master, msdb and tempdb (Scored)	☐	☐
3.3	Drop Orphaned Users From SQL Server Databases (Scored)	☐	☐
3.4	Do not use SQL Authentication in contained databases (Scored)	☐	☐
4	**Password Policies**		
4.1	Set the 'MUST_CHANGE' Option to ON for All SQL Authenticated Logins (Not Scored)	☐	☐
4.2	Set the 'CHECK_EXPIRATION' Option to ON for All SQL Authenticated Logins Within the Sysadmin Role (Scored)	☐	☐
4.3	Set the 'CHECK_POLICY' Option to ON for All SQL Authenticated Logins (Scored)	☐	☐
5	**Auditing and Logging**		
5.1	Set the 'Maximum number of error log files' setting to greater than or equal to 12 (Not Scored)	☐	☐
5.2	Set the 'Default Trace Enabled' Server Configuration Option to 1 (Scored)	☐	☐
5.3	Set 'Login Auditing' to failed logins (Not Scored)	☐	☐
5.4	Use SQL Server Audit to capture both failed and successful logins (Not Scored)	☐	☐
6	**Application Development**		

6.1	Sanitize Database and Application User Input (Not Scored)	☐	☐
6.2	Set the 'CLR Assembly Permission Set' to SAFE_ACCESS for All CLR Assemblies (Scored)	☐	☐
7	**Encryption**		
7.1	Ensure Symmetric Key encryption algorithm is AES_128 or higher in non-system databases (Scored)	☐	☐
7.2	Ensure asymmetric key size is greater than or equal to 2048 in non-system databases (Scored)	☐	☐
8	**Appendix: Additional Considerations**		
8.1	SQL Server Browser Service (Not Scored)	☐	☐

Appendix: Change History

Date	Version	Changes for this version
12-17-2014	1.0.0	Initial Release