

Security Configuration Benchmark For

Microsoft SQL Server 2008

Version 1.0.0-A9

June 29, 2012

Copyright 2001-2012, The Center for Internet Security

<http://cisecurity.org>

feedback@cisecurity.org

Background.

CIS provides benchmarks, scoring tools, software, data, information, suggestions, ideas, and other services and materials from the CIS website or elsewhere ("**Products**") as a public service to Internet users worldwide. Recommendations contained in the Products ("**Recommendations**") result from a consensus-building process that involves many security experts and are generally generic in nature. The Recommendations are intended to provide helpful information to organizations attempting to evaluate or improve the security of their networks, systems and devices. Proper use of the Recommendations requires careful analysis and adaptation to specific user requirements. The Recommendations are not in any way intended to be a "quick fix" for anyone's information security needs.

No representations, warranties and covenants.

CIS makes no representations, warranties or covenants whatsoever as to (i) the positive or negative effect of the Products or the Recommendations on the operation or the security of any particular network, computer system, network device, software, hardware, or any component of any of the foregoing or (ii) the accuracy, reliability, timeliness or completeness of any Product or Recommendation. CIS is providing the Products and the Recommendations "as is" and "as available" without representations, warranties or covenants of any kind.

User agreements.

By using the Products and/or the Recommendations, I and/or my organization ("**we**") agree and acknowledge that:

No network, system, device, hardware, software or component can be made fully secure;
We are using the Products and the Recommendations solely at our own risk;

We are not compensating CIS to assume any liabilities associated with our use of the Products or the Recommendations, even risks that result from CIS's negligence or failure to perform;

We have the sole responsibility to evaluate the risks and benefits of the Products and Recommendations to us and to adapt the Products and the Recommendations to our particular circumstances and requirements;

Neither CIS, nor any CIS Party (defined below) has any responsibility to make any corrections, updates, upgrades or bug fixes or to notify us if it chooses at its sole option to do so; and

Neither CIS nor any CIS Party has or will have any liability to us whatsoever (whether based in contract, tort, strict liability or otherwise) for any direct, indirect, incidental, consequential, or special damages (including without limitation loss of profits, loss of sales, loss of or damage to reputation, loss of customers, loss of software, data, information or emails, loss of privacy, loss of use of any computer or other equipment, business interruption, wasted management or other staff resources or claims of any kind against us from third parties) arising out of or in any way connected with our use of or our inability to use any of the Products or Recommendations (even if CIS has been advised of the possibility of such damages), including without limitation any liability associated with infringement of intellectual property, defects, bugs, errors, omissions, viruses, worms, backdoors, Trojan horses or other harmful items.

Grant of limited rights.

CIS hereby grants each user the following rights, but only so long as the user complies with all of the terms of these Agreed Terms of Use:

Except to the extent that we may have received additional authorization pursuant to a written agreement with CIS, each user may download, install and use each of the Products on a single computer;

Each user may print one or more copies of any Product or any component of a Product that is in a .txt, .pdf, .doc, .mcw, or .rtf format, provided that all such copies are printed in full and are kept intact, including without limitation the text of this Agreed Terms of Use in its entirety.

Retention of intellectual property rights; limitations on distribution.

The Products are protected by copyright and other intellectual property laws and by international treaties. We acknowledge and agree that we are not acquiring title to any intellectual property rights in the Products and that full title and all ownership rights to the Products will remain the exclusive property of CIS or CIS Parties. CIS reserves all rights not expressly granted to users in the preceding section entitled "Grant of limited rights." Subject to the paragraph entitled "Special Rules" (which includes a waiver, granted to some classes of CIS Members, of certain limitations in this paragraph), and except as we may have otherwise agreed in a written agreement with CIS, we agree that we will not (i) decompile, disassemble, reverse engineer, or otherwise attempt to derive the source code for any software Product that is not already in the form of source code; (ii) distribute, redistribute, encumber, sell, rent, lease, lend, sublicense, or otherwise transfer or exploit rights to any Product or any component of a Product; (iii) post any Product or any component of a Product on any website, bulletin board, ftp server, newsgroup, or other similar mechanism or device, without regard to whether such mechanism or device is internal or external, (iv) remove or alter trademark, logo, copyright or other proprietary notices, legends, symbols or labels in any Product or any component of a Product; (v) remove these Agreed Terms of Use from, or alter these Agreed Terms of Use as they appear in, any Product or any component of a Product; (vi) use any Product or any component of a Product with any derivative works based directly on a Product or any component of a Product; (vii) use any Product or any component of a Product with other products or applications that are directly and specifically dependent on such Product or any component for any part of their functionality, or (viii) represent or claim a particular level of compliance with a CIS Benchmark, scoring tool or other Product. We will not facilitate or otherwise aid other individuals or entities in any of the activities listed in this paragraph.

We hereby agree to indemnify, defend and hold CIS and all of its officers, directors, members, contributors, employees, authors, developers, agents, affiliates, licensors, information and service providers, software suppliers, hardware suppliers, and all other persons who aided CIS in the creation, development or maintenance of the Products or Recommendations ("CIS Parties") harmless from and against any and all liability, losses, costs and expenses (including attorneys' fees and court costs) incurred by CIS or any CIS Party in connection with any claim arising out of any violation by us of the preceding paragraph, including without limitation CIS's right, at our expense, to assume the exclusive defense and control of any matter subject to this indemnification, and in such case, we agree to cooperate with CIS in its defense of such claim. We further agree that all CIS Parties are third-party beneficiaries of our undertakings in these Agreed Terms of Use.

Special rules.

CIS has created and will from time to time create special rules for its members and for other persons and organizations with which CIS has a written contractual relationship. Those special rules will override and supersede these Agreed Terms of Use with respect to the users who are covered by the special rules. CIS hereby grants each CIS Security Consulting or Software Vendor Member and each CIS Organizational User Member, but only so long as such Member remains in good standing with CIS and complies with all of the terms of these Agreed Terms of Use, the right to distribute the Products and Recommendations within such Member's own organization, whether by manual or electronic means. Each such Member acknowledges and agrees that the foregoing grant is subject to the terms of such Member's membership arrangement with CIS and may, therefore, be modified or terminated by CIS at any time.

Choice of law; jurisdiction; venue.

We acknowledge and agree that these Agreed Terms of Use will be governed by and construed in accordance with the laws of the State of Maryland, that any action at law or in equity arising out of or relating to these Agreed Terms of Use shall be filed only in the courts located in the State of Maryland, that we hereby consent and submit to the personal jurisdiction of such courts for the purposes of litigating any such action. If any of these Agreed Terms of Use shall be determined to be unlawful, void, or for any reason unenforceable, then such terms shall be deemed severable and shall not affect the validity and enforceability of any remaining provisions. We acknowledge and agree that we have read these Agreed Terms of Use in their entirety, understand them and agree to be bound by them in all respects.

Table of Contents

Table of Contents	4
Overview	10
Consensus Guidance	10
Intended Audience	10
Acknowledgements.....	10
Typographic Conventions.....	11
Configuration Levels	11
Level-I Benchmark settings/actions	11
Level-II Benchmark settings/actions.....	11
Scoring Status	11
Scorable	11
Not Scorable	12
1. Recommendations.....	12
1.1 Physical and Network Security (Environment).....	12
1.1.1 Access to SQL Server (Level 1, Not Scorable)	12
1.1.2 Domain Environment (Level 1, Scorable)	12
1.1.3 Test and Development Servers (Level 1, Not Scorable).....	13
1.1.4 Dedicated Server (Level 1, Not Scorable).....	14
1.1.5 Disable Unused Network Protocols (Level 1, Scorable).....	14
1.2 Operating System Level Configuration	15
1.2.1 Operating System Choice (Level 1, Scorable).....	15
1.2.2 Disk Subsystem (Level 1, Not Scorable).....	15
1.2.3 Separate Partitions For Database Files (Level 1, Scorable).....	16
1.2.4 Separate Partition For Backups (Level 1, Scorable).....	16
1.2.5 Separate tempdb Partition (Level 1, Scorable).....	17
1.2.6 Volume / partition type (Level 1, Scorable).....	18
1.2.7 Windows 2008 Server Benchmark (Level 1, Scorable).....	18
1.2.8 Group Policy (Level 1, Scorable).....	19
1.2.9 Internet Explorer Enhanced Security Configuration (IE ESC) (Level 1, Scorable)	20
1.2.10 Permissions on OS tools	20
1.3 Operating System Services	21
1.3.1 Application Information (Appinfo) (Level 1, Scorable).....	21
1.3.2 Application Layer Gateway (ALG) (Level 1, Scorable).....	22
1.3.3 ASP.NET State Service (aspnet_state) (Level 1, Scorable).....	22
1.3.4 Computer Browser (browser) (Level 1, Scorable)	23
1.3.5 Distributed Transaction Coordinator (MsDtc) (Level 1, Scorable).....	23
1.3.6 Function Discovery Provider Host (fdpHost) (Level 1, Scorable).....	24
1.3.7 Function Discovery Resource Publication (FDResPub) (Level 1, Scorable)....	24
1.3.8 Internet Connection Sharing (ICS) (SharedAccess) (Level 1, Scorable).....	25
1.3.9 Link-Layer Topology Discovery Mapper (lltdsvc) (Level 1, Scorable).....	25
1.3.10 Multimedia Class Scheduler (MMCSS) (Level 1, Scorable).....	26
1.3.11 Print Spooler (Spooler) (Level 1, Scorable).....	26

1.3.12	Remote Access Auto Connection Manager (RasAuto) (Level 1, Scorable) ...	27
1.3.13	Remote Access Connection Manager (RasMan) (Level 1, Scorable).....	27
1.3.14	Remote Registry (RemoteRegistry) (Level 2, Scorable).....	28
1.3.15	Routing and Remote Access (RemoteAccess) (Level 1, Scorable)	28
1.3.16	Telephony (TapiSrv) (Level 1, Scorable)	29
1.3.17	Windows Audio (AudioSrv) (Level 1, Scorable).....	29
1.3.18	Windows Audio Endpoint Builder (AudioEndpointBuilder) (Level 1, Scorable) 30	
1.3.19	Windows Color System (WcsPlugInService) (Level 1, Scorable).....	30
1.3.20	Windows Font Cache Service (FontCache) (Level 1, Scorable).....	31
1.3.21	Windows Remote Management (WS-Management) (WinRM) (Level 2, Scorable) 31	
1.3.22	WinHTTP Web Proxy Auto-Discovery Service (WinHttpAutoProxySvc) (Level 1, Scorable).....	32
1.3.23	Distributed Link Tracking Client (TrkWks) (Level 1, Scorable)	33
1.3.24	Windows Installer (msiserver) (Level 1, Scorable).....	33
1.4	Service Account Management	34
1.4.1	Dedicated SQL Server Service Account (Level 1, Not Scorable)	34
1.4.2	Local users group membership (Level 1, Scorable)	34
1.4.3	Domain service account group membership (Level 1, Not Scorable).....	35
1.4.4	Service Account Dedicated to the SQL Server Instance (Level 2, Not Scorable) 35	
1.4.5	Different Service Accounts (Level 2, Scorable).....	36
1.5	SQL Server Service Account Rights.....	37
1.5.1	MSSQL Server Service Account (Level 1, Not Scorable)	37
1.5.2	Grant the SQL Server Service Account Only Necessary Privileges (Level 1, Scorable)	37
1.5.3	Lock Pages in Memory Privilege (Level 1, Scorable)	38
1.5.4	Perform Volume Maintenance Tasks Privilege (Level 2, Scorable)	39
1.5.5	Password Management (Level 1, Scorable)	40
1.6	SQL Server Agent Service Account Rights	41
1.6.1	SQL Server Agent Service Account (Level 1, Scorable).....	41
1.6.2	Grant the SQL Server Agent Account Necessary Privileges (Level 1, Scorable) 41	
1.6.3	SQL Server Agent Proxy Accounts (Level 1, Not Scorable).....	42
1.7	Integration Services service account rights.....	43
1.7.1	SQL Server Integration Services Service Account (Level 1, Scorable)	43
1.7.2	Grant the SQL Server Integration Services Service Account Only Necessary Privileges (Level 1, Scorable).....	43
1.8	SQL Server Installation and Patches	44
1.8.1	Domain Controller Installation (Level1, Scorable)	44
1.8.2	Service Packs and Hotfixes (Level 1, Not Scorable).....	45
1.8.3	SQL Server Agent Service Startup (Level 1, Scorable)	45
1.8.4	Instance Naming Conventions (Level 1, Not Scorable).....	46
1.8.5	Inventory of SQL Server Instances (Level 1, Not Scorable)	46
1.8.6	Sample databases (Level 1, Scorable)	47

1.8.7	Lock Down sa Account (Level 1, Scorable)	48
1.8.8	Strong sysadmin password (Level 1, Not Scorable)	48
1.8.9	SQL Server Program Directory Permissions (Level 1, Scorable)	49
1.8.10	SQL Server database instance directory permissions (Level 1, Scorable)	49
1.8.11	SQL Server Data Directory Permissions (Level 1, Scorable)	50
1.8.12	SQL Server Log Directory Permissions (Level 1, Scorable)	51
1.8.13	Monitor SQL Server (Level 1, Scorable)	52
1.9	SQL Server Interface Configuration	52
1.9.1	Disable Unnecessary Protocols (Level 1, Scorable)	52
1.9.2	SQL Server Ports (Level 1, Scorable)	53
1.9.3	Limit Network Interfaces (Level 1, Scorable)	53
1.9.4	Dedicated Admin Connection (DAC) Port (Level 1, Scorable)	54
1.9.5	Disable SQL Server Browser Service (Level 1, Scorable)	54
1.9.6	Hide SQL Server Instance (Level 2, Scorable)	55
1.9.7	Encryption (Level 2, Scorable)	56
1.9.8	Disable Native XML Web Services (Level 1, Scorable)	57
1.9.9	Database Mirroring Encryption (Level 1, Scorable)	57
1.10	SQL Server Configuration Options	58
1.10.1	Ad Hoc Distributed Queries (Level 1, Scorable)	58
1.10.2	C2 audit mode (Level 2, Scorable)	59
1.10.3	CLR enabled (Level 1, Scorable)	59
1.10.4	Common criteria compliance enabled (Level 1, Scorable)	60
1.10.5	Cross db ownership chaining (Level 1, Scorable)	61
1.10.6	Database Mail XPs (Level 1, Scorable)	61
1.10.7	Default trace enabled (Level 1, Scorable)	62
1.10.8	Maximum Server Memory (Level 1, Scorable)	63
1.10.9	Ole Automation Procedures (Level 1, Scorable)	64
1.10.10	Remote Access (Level 1, Scorable)	64
1.10.11	Remote admin connections (Level 1, Scorable)	65
1.10.12	Scan for startup procs (Level 1, Scorable)	66
1.10.13	SQL Mail XPs (Level 1, Scorable)	67
1.10.14	Xp_cmdshell (Level 1, Scorable)	67
1.11	General SQL Server Server Level Settings	68
1.11.1	Database Mail Access (Level 1, Scorable)	68
1.11.2	Trace Messages (Level 1, Scorable)	69
1.11.3	SQL Server Event Forwarding (Level 2, Not Scorable)	69
1.11.4	Audit Login Attempts (Level 1, Scorable)	70
1.11.5	Do Not Use Remote Servers (Level 1, Scorable)	71
1.11.6	Linked Servers (Level 1, Scorable)	71
1.11.7	sp_processmail (Level 1, Scorable)	72
1.12	SQL Server Database Level Settings	73
1.12.1	Encrypt Stored Procedure Definition (Level 2, Not Scorable)	73
1.12.2	DDL Permissions (Level 1, Scorable)	73
1.12.3	Limit Guest User Permissions (Level 1, Scorable)	74
1.12.4	Remove Orphan Users (Level 1, Scorable)	75
1.12.5	Disable Auto Close Option (Level 1, Scorable)	75

1.12.6	Use Transparent Data Encryption (TDE) (Level 2, Scorable).....	76
1.12.7	Turn Off Trustworthy Option (Level 2, Scorable).....	76
1.13	Extended stored procedures	77
1.13.1	User-defined extended stored procedures (Level 1, Scorable).....	77
1.13.2	xp_availablemedia (Level 1, Scorable)	78
1.13.3	xp_cmdshell (Level 1, Scorable)	78
1.13.4	xp_dirtree (Level 1, Scorable).....	79
1.13.5	xp_enumgroups (Level 1, Scorable).....	80
1.13.6	xp_fixeddrives (Level 1, Scorable).....	81
1.13.7	xp_getnetname (Level 1, Scorable)	81
1.13.8	xp_logevent (Level 1, Scorable)	82
1.13.9	xp_loginconfig (Level 1, Scorable).....	83
1.13.10	xp_msver (Level 1, Scorable)	83
1.13.11	xp_readerrorlog (Level 1, Scorable)	84
1.13.12	xp_servicecontrol (Level 1, Scorable)	85
1.13.13	xp_sprintf (Level 1, Scorable)	86
1.13.14	xp_sscanf (Level 1, Scorable)	87
1.13.15	xp_subdirs (Level 1, Scorable).....	87
1.13.16	xp_regaddmultistring (Level 1, Scorable).....	88
1.13.17	xp_regdeletekey (Level 1, Scorable)	89
1.13.18	xp_regdeletevalue (Level 1, Scorable).....	89
1.13.19	xp_regenumvalues (Level 1, Scorable).....	90
1.13.20	xp_regremovemultistring (Level 1, Scorable)	91
1.13.21	xp_regwrite (Level 1, Scorable)	91
1.13.22	xp_regread (Level 1, Scorable).....	92
1.14	Authentication and Authorization.....	93
1.14.1	Authentication mode (Level 1, Scorable).....	93
1.14.2	Shared Accounts (Level 1, Scorable).....	94
1.14.3	Windows Logins (Level 1, Scorable)	94
1.14.4	SQL Logins (Level 1, Not Scorable).....	94
1.14.5	Assigning System Administrators role (Level 1, Not Scorable)	95
1.14.6	Ensure SQL Logins have a Strong Password (Level 1, Not Scorable)	96
1.14.7	Fixed Server Roles (Level 1, Not Scorable)	96
1.14.8	Database Roles (Level 1, Scorable)	97
1.14.9	Guest Database User (Level 1, Scorable).....	98
1.14.10	Using the GRANT option (Level 1, Scorable).....	98
1.14.11	User-defined Database Roles (Level 2, Not Scorable).....	99
1.14.12	Role Assignments (Level 1, Not Scorable).....	100
1.15	Auditing and Logging	100
1.15.1	Review Collected Audit Log (Level 1, Not Scorable)	100
1.15.2	Secure Audit Files (Level 1, Not Scorable)	101
1.15.3	SQL Server Logs (Level 1, Scorable).....	101
1.15.4	Review SQL Permissions Granted (Level 1, Scorable).....	102
1.15.5	Periodic scan of stored procedures (Level 1, Not Scorable).....	102
1.16	Audit SQL Trace Configuration.....	103
1.16.1	SQL Trace Setup (Level 1, Scorable).....	103

1.16.2	Audit Add Login to Server Role Event Class (Level 1, Scorable)	104
1.16.3	Audit Add Member to DB Role Event Class (Level 1, Scorable)	105
1.16.4	Audit App Role Change Password Event Class (Level 1, Scorable)	106
1.16.5	Audit Backup/Restore Event Class (Level 1, Scorable).....	107
1.16.6	Audit Change Audit Event Class (Level 1, Scorable).....	108
1.16.7	Audit Change Database Owner Event Class (Level 1, Scorable)	109
1.16.8	Audit Database Management Event Class (Level 1, Scorable).....	110
1.16.9	Audit Database Mirroring Login Event Class (Level 1, Scorable).....	111
1.16.10	Audit Database Object Access Event Class (Level 2, Scorable).....	112
1.16.11	Audit Database Object GDR Event Class (Level 1, Scorable)	113
1.16.12	Audit Database Object Management Event Class (Level 1, Scorable)	114
1.16.13	Audit Database Object Take Ownership Event Class (Level 1, Scorable)....	115
1.16.14	Audit Database Principal Impersonation Event Class (Level 1, Scorable)..	116
1.16.15	Audit Database Principal Management Event Class (Level 1, Scorable).....	117
1.16.16	Audit Database Scope GDR Event Class (Level 1, Scorable).....	118
1.16.17	Audit DBCC Event Class (Level 1, Scorable)	119
1.16.18	Audit Login Change Password Event Class (Level 1, Scorable).....	120
1.16.19	Audit Login Change Property Event Class (Level 1, Scorable)	121
1.16.20	Audit Login Event Class (Level 1, Scorable)	122
1.16.21	Audit Login Failed Event Class (Level 1, Scorable)	123
1.16.22	Audit Logout Event Class (Level 1, Scorable)	124
1.16.23	Audit Schema Object Access Event Class (Level 2, Scorable).....	125
1.16.24	Audit Schema Object GDR Event Class (Level 1, Scorable).....	126
1.16.25	Audit Schema Object Take Ownership Event Class (Level 1, Scorable).....	127
1.16.26	Audit Server Alter Trace Event Class (Level 1, Scorable)	128
1.16.27	Audit Server Object GDR Event Class (Level 1, Scorable)	129
1.16.28	Audit Server Object Management Event Class (Level 1, Scorable).....	130
1.16.29	Audit Server Object Take Ownership Event Class (Level 1, Scorable)	131
1.16.30	Audit Server Operation Event Class (Level 1, Scorable).....	132
1.16.31	Audit Server Principal Impersonation Event Class (Level 1, Scorable)	133
1.16.32	Audit Server Principal Management Event Class (Level 1, Scorable)	134
1.16.33	Audit Server Scope GDR Event Class (Level 1, Scorable)	135
1.16.34	Audit Server Starts and Stops Event Class (Level 1, Scorable).....	136
1.16.35	Audit Database Operation Event Class (Level 1, Scorable)	137
1.16.36	Audit Schema Object Management Event Class (Level 1, Scorable)	138
1.17	Backup and Disaster Recovery	139
1.17.1	Backup Strategy (Level 1, Not Scorable).....	139
1.17.2	Take Regular Full Backups (Level 1, Scorable).....	140
1.17.3	Backing up Master Database (Level 1, Not Scorable).....	140
1.17.4	Backing up MSDB database (Level 1, Not Scorable).....	141
1.17.5	Restrict access to backup files to System Administrators (Level 1, Scorable) 142	
1.17.6	Off-server Backup (Level 1, Not Scorable)	142
1.17.7	Off-site Backup (Level 2, Not Scorable)	143
1.17.8	Encrypt Backups (Level 2, Not Scorable).....	143
1.17.9	Regular Restore Test (Level 1, Not Scorable)	143

1.17.10	Regular DBCC CHECKDB (Level 1, Scorable).....	144
1.17.11	Encryption Components Backup (Level 1, Scorable).....	144
1.18	Replication	145
1.18.1	Replication Enabled (Level 1, Scorable).....	145
1.18.2	Replication administration roles (Level 1, Not Scorable).....	146
1.18.3	Replication Agent Account (Level 2, Scorable)	146
1.18.4	Secure Replication Snapshot Folder (Level 1, Scorable).....	147
1.18.5	Secure Replication Over the Internet (Level 1, Not Scorable)	148
1.18.6	Filtering Replication (Level 1,Not Scorable)	148
1.18.7	Securing Distribution Databases (Level 1, Not Scorable)	149
1.18.8	Securing Subscription Databases (Level 1, Not Scorable).....	149
1.19	Application Development	150
1.19.1	Use Recent Driver (Level 1, Not Scorable)	150
1.19.2	Error Handling (Level 1, Not Scorable).....	150
1.19.3	Sanitize User Input (Level 1, Not Scorable)	151
1.19.4	Customer Awareness to SQL Express (Level 1, Not Scorable).....	151
1.19.5	CLR UNSAFE_ACCESS Permission Set (Level 1, Scorable).....	152
1.19.6	CLR SAFE_ACCESS Permission Set (Level 2, Scorable)	153
1.19.7	Plain Text Password (Level 1, Scorable).....	153
1.20	TBD: DROPPED SECTIONS/RECOMMENDATIONS	154
1.20.1	Permission to Start SQL Writer Service (Level 1, Scorable)	154
1.20.2	Permission to read the Remote Procedure Call service (Level 1, Scorable) 154	
1.20.3	Permission to read the Event Log service (Level 1, Scorable).....	155
1.20.4	Permission to Start SQL Server Active Directory Helper Service (Level 1, Scorable) 155	
1.20.5	Web Task Extended Stored procedures	156
1.20.6	SQL Mail Extended Procedures.....	156
1.20.7	OLE Automation Stored Procedures	156
Appendix A: References.....		156
Appendix B: Change History.....		157

Overview

This document, *Security Configuration Benchmark for Microsoft SQL Server 2008*, provides prescriptive guidance for establishing a secure configuration posture for *Microsoft SQL Server 2008* versions <Target Version> – <Target Name> running on *Microsoft Windows Server 2008 R2*. This guide was tested against *Microsoft SQL Server 2008 64-bit version 10.00.5500 (Service Pack 3)* as installed by Microsoft SQL Server 2008 Service Pack 3. To obtain the latest version of this guide, please visit <http://cisecurity.org>. If you have questions, comments, or have identified ways to improve this guide, please write us at feedback@cisecurity.org.

Consensus Guidance

This benchmark was created using a consensus review process comprised of volunteer and contract 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 released to the public Internet. During this phase, all feedback provided by the Internet community is reviewed by the consensus team for incorporation in to the CIS benchmark. If you are interested in participating in the consensus review process, please send us a note to 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 2008* on a Microsoft Windows platform.

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:

Authors

Erik Bitemo

Maintainers

Editors

Testers

Contributors and Reviewers

Typographic 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

Configuration Levels

This section defines the configuration levels that are associated with each benchmark recommendation. Configuration levels represent increasing levels of security assurance.

Level-I Benchmark settings/actions

Level-I Benchmark recommendations are intended to:

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

Level-II Benchmark settings/actions

Level-II Benchmark recommendations exhibit one or more of the following characteristics:

- are intended for environments or use cases where security is paramount
- acts as defense in depth measure
- may negatively inhibit the utility or performance of the technology

Scoring Status

This section defines the scoring statuses used within this document. The scoring status indicates whether compliance with the given recommendation is discernable in an automated manner.

Scorable

The platform's compliance with the given recommendation can be determined via automated means.

Not Scorable

The platform's compliance with the given recommendation cannot be determined via automated means.

1. Recommendations

Comment [BEG1]: Note that all recommendation is labeled as level 1, scorable in this phase.

1.1 Physical and Network Security (Environment)

Majority of physical and network security recommendations are not specific to SQL Server – e.g. place your server to a physically secure location or establish a password management system. Those are not shown up here.

Comment [BEG2]: Do you agree with this approach?

1.1.1 Access to SQL Server (Level 1, Not Scorable)

Description:

Limit access to local network, don't face SQL Server to the Internet.

Rationale:

Exposing SQL Server to the Internet implies a high security risk.

Remediation:

Ensure that your SQL Server is not available through the Internet. If you can't avoid installing SQL Server onto a server exposed to the Internet, configure it to listen on a non-internet facing IP address.

Audit:

Try to connect to the SQL Server instance via the public IP address of the server if it has one. The connection attempt must fail on the network level.

Default Value:

By default, SQL Server is not listening on TCP. After activating TCP/IP, it binds to all interfaces and IP addresses.

References:

1.1.2 Domain Environment (Level 1, Scorable)

Description:

Make the underlying OS part of a domain if possible.

Rationale:

A domain environment helps enforcing security policies on computers in a secure and efficient way.

Remediation:

Join the computer a domain.

Audit:

Verify if the computer is part of a domain.

Default Value:

A computer is not part of a domain by default.

References:

1.1.3 Test and Development Servers (Level 1, Not Scorable)

Description:

Maintain test and development server on a separate network segment from the production servers and ensure that application and database servers in an environment can't talk to servers in another environment.

Rationale:

Test and development environments more open to the office and public network than production servers usually. This increases the risk of malicious attacks. Bugs or viruses that may occur in the lower environments should never have the opportunity to infiltrate to the higher environments. Also ensure that hot fixes or service packs are thoroughly tested in lower environments before promoting to production.

Remediation:

Keep test and development servers on a separate network segment from production servers. If this is not possible, disable any linked server connections on the servers.

Audit:

Validate that no network connection exists between test and production servers by running the following command line script:

```
C:\>telnet <servername.fqdn> 1433
```

The connection attempt must time out.

Or

Evaluate all linked servers in non-production environments to see if they can connect to the production machines.

Run the following code snippet to determine all linked servers on a given SQL Server instance:

```
SELECT *  
FROM sys.linked_logins  
WHERE server_id != 0
```

Default Value:

By default, SQL Server can't talk to other instances; however, they're available by application running side by side by SQL Server on the same network

References:

1.1.4 Dedicated Server (Level 1, Not Scorable)

Description:

Install SQL Server on a computer that does not provide additional services, e.g., web or mail services.

Rationale:

Vulnerabilities in other application services could lead to a compromise of the SQL Server. Performance issues may also occur when SQL Server shared resources with other applications.

Remediation:

Ensure that the SQL Server instance is on its own dedicated machine that does not run external applications such as a web server or a mail server.

Audit:

Analyze the programs currently installed on the SQL Server machine. If other applications reside on the machine contends for SQL Server resources, consider moving those applications or the SQL Server instance to a different machine.

Default Value:

Not applicable.

References:

1.1.5 Disable Unused Network Protocols (Level 1, Scorable)

Description:

Disable network protocols not in use in the operating system.

Rationale:

Leaving unused network protocols enabled exposes the computer to more attacks and consumes additional resources.

Remediation:

1. Open `ncpa.cpl` in the Run dialog in Start menu.
2. Open the first interface in Network Connections by double-clicking on it.
3. Check the list of items checked in the `This connection uses the following items` section.
4. Deselect the items not necessary.
5. Click OK.
6. Repeat it for all interfaces.

Audit:

1. Open `ncpa.cpl` in the Run dialog in Start menu.
2. Open the first interface in Network Connections by double-clicking on it.

3. Check the list of items checked in the `This connection uses the following items` section.
4. Ensure they're all necessary.
5. Repeat it for all interfaces.

Default Value:

By default, installed network protocols are active on network interfaces.

References:

1.2 Operating System Level Configuration

1.2.1 Operating System Choice (Level 1, Scorable)

Description:

Use the most recent available OS supported by the product. This is Windows Server 2008 (or 2008 R2) for SQL Server 2008.

Rationale:

Using the latest supported operating system allows using the full potential of the SQL Server product. In general, later editions of the operating system are more robust, thus enhancing stability.

Remediation:

Install SQL Server onto a server running Windows Server 2008 R2.

Audit:

Verify the version of the operating system by running the following command in a command prompt:

```
C:\>ver
```

The output should return version 6.x.

Default Value:

Not Applicable.

References:

1.2.2 Disk Subsystem (Level 1, Not Scorable)

Description:

Use RAID for volumes holding operating system and data files.

Rationale:

Using RAID increases availability and security from data loss.

Remediation:

Configure your hard drives to use RAID.

Audit:

If not currently known, investigate the current disk subsystem to determine the RAID level. Consult your hardware guide for RAID level specifics.

Default Value:

Not Applicable

References:

1.2.3 Separate Partitions For Database Files (Level 1, Scorable)

Description:

Create separate partitions for OS/SQL program files, SQL data files and SQL transaction logs.

Rationale:

Separating backup files to different partition minimizes the impact of a possible disk full event and allows more adequate monitoring. Note that separating log files into a dedicated partition will cause a significant load onto the disk under high traffic.

Remediation:

Configure SQL Server to use different partitions for user data, user log and default backup directories. Move any already existing files to the appropriate partitions.

Audit:

Verify if existing database files are on separate partitions by running the following T-SQL query:

```
SELECT db_name(database_id) AS db, name, data_space_id, physical_name
FROM sys.master_files
```

File paths in the physical name column must use different partitions for files with data_space_id = 0 (log files) than the others. No files should be located on the system partition.

Default Value:

By default, all files are stored under the SQL Server program files directory, which is on the system partition.

References:

1.2.4 Separate Partition For Backups (Level 1, Scorable)

Description:

Create separate partitions for database backup files, if taken to disk.

Rationale:

Separating backup files to different partition minimizes the impact of any issue regarding the backup system resulting in more disk space occupied than usually.

Remediation:

Configure SQL Server to use different partition for default backup directory. Move any already existing backup files to the appropriate partition.

Audit:

Verify if existing database files are on separate partitions by running the following T-SQL query:

```
SELECT db_name(database_id) AS db, name, data_space_id, physical_name
FROM sys.master_files
```

File paths in the physical name column must use different partitions for files with data_space_id = 0 (log files) than the others. No files should be located on the system partition.

Default Value:

By default, all database files and backup files are stored under the SQL Server program files directory, which is on the system partition.

References:

1.2.5 Separate tempdb Partition (Level 1, Scorable)

Description:

Create separate partition for tempdb.

Rationale:

Tempdb is used as an on-disk storage for temp tables and complex sort and join operations not fitting in memory. As such, it might grow to a significant size. Separating it can prevent the other database files from a disk full event.

Remediation:

Configure SQL Server to store tempdb files on a separate partition from other operating system and SQL files by running the following T-SQL query (replace filename parameters with the appropriate ones):

```
ALTER DATABASE tempdb MODIFY FILE (NAME = tempdev, FILENAME =
'T:\mssql\Data\tempdb.mdf')
ALTER DATABASE tempdb MODIFY FILE (NAME = templog, FILENAME =
'T:\mssql\Data\templog.ldf')
```

Restart SQL Server.

Audit:

Verify if tempdb database files are on separate partitions by running the following T-SQL query:

```
SELECT db_name(database_id) AS db, name, data_space_id, physical_name
FROM sys.master_files
```

File paths in the physical name column must use different partitions for tempdb files than for other files. The partition for tempdb files should not be the system partition.

Default Value:

By default, all database files, including tempdb files are stored under the SQL Server program files directory, which is on the system partition.

References:

1.2.6 Volume / partition type (Level 1, Scorable)

Description:

Format all volumes with the NTFS file system.

Rationale:

NTFS allows assigning rights to specific users or groups of users. The more granular and restricted your security scheme, the more secure your installation will likely be.

Remediation:

Consider using the NTFS file system for all production level servers if you are not currently doing so.

Audit:

To determine if your drives are currently NTFS formatted, perform the following steps:

1. Open Computer Manager by running `compmgmt.msc`.
2. Expand the Storage dropdown.
3. Navigate to Disk Management to view the drives. The format will be listed with the drive.

Default Value:

NTFS is the default choice when installing the operating system.

References:

1.2.7 Windows 2008 Server Benchmark (Level 1, Scorable)

Description:

Configure Windows 2008 Server Benchmark (v1.1.0) Enterprise Security Profile recommendations with the additions/amendments detailed in this benchmark, especially in section 1.2.

Rationale:

Proper security configuration of the operating system provides a solid environment for SQL Server to run.

Remediation:

Follow the recommendations of the Windows 2008 Server Benchmark.

Audit:

Follow the audit sections of the Windows 2008 Server Benchmark recommendations.

Default Value:

Not Applicable

References:

1.2.8 Group Policy (Level 1, Scorable)

Description:

If Group Policy is used to enforce benchmark settings, make sure that User Rights Assignment settings are not enforced by Group Policy.

Rationale:

Several privileges under Local Policies\User Rights Assignment are required for SQL Server service accounts. Enforcing the Windows 2008 Server Benchmark recommendations via Group Policy results in either failure to run SQL Server or granting excessive privileges to the SQL Server service accounts.

Remediation:

Ensure that no Group Policy is set for the following subtree on systems running SQL Server:

```
Computer Configuration\Windows Settings\Security Settings\Local Policies\User Rights Assignment
```

Audit:

1. Open Local Security Policy by running `secpol.msc`.
2. Browse to Local Policies\User Rights Assignment.
3. Check if no policy on the right side comes from Group Policy (has a script icon).

Default Value:

No Group Policy is set for this subtree by default.

References:

1.2.9 Internet Explorer Enhanced Security Configuration (IE ESC) (Level 1, Scorable)

Description:

IE ESC provides a security layer when Internet Explorer is used on the server.

Rationale:

IE ESC ensures that no unauthorized actions can be done to the system running the browser application while browsing websites. Nevertheless, it is recommended to not browse any websites from the server if possible.

Remediation:

1. Close all instances of Internet Explorer.
2. Click Start, point to Administrative Tools, and then click Server Manager.
3. If a User Account Control dialog box appears, click Continue.
4. Under Security Summary, click Configure IE ESC.
5. Under Administrators, click On (Recommended).
6. Under Users, click On (Recommended).
7. Click OK.

Audit:

1. Click Start, point to Administrative Tools, and then click Server Manager.
2. If a User Account Control dialog box appears, click Continue.
3. Under Security Summary, click Configure IE ESC.
4. Check if under both Administrators and Users, On (Recommended) is selected.
5. Click Cancel.

Default Value:

Administrators: On, Users: On

References:

1.2.10 Permissions on OS tools

Description:

Restrict access to the executables in the System32 directory eg. `Explorer.exe` and `cmd.exe`.

Rationale:

The majority of the operating systems tools are deployed in the system32 directory in the C drive. Ensure that only the necessary accounts have permission to this folder.

Remediation:

Comment [BEG3]: This likely should be broken down to separate admin and users?

Comment [b4]: Is it beneficial really? A copied cmd.exe works fine.

Remove the Users group's permission (if any) from the executables. Assign Administrators Full Control.

Audit:

Review permission on the following folders in the Windows OS:

%SystemRoot%\System32
%SystemRoot%\System

Default Value:

By default this folder is hidden.

References:

<http://support.microsoft.com/kb/307545>

1.3 Operating System Services

Disable the following services on your Windows Server 2008 (R2) installation unless they're necessary for a particular application. For configuration of Windows Server 2003 services, please refer to the SQL Server 2005 version 2.0 Benchmark.

Comment [BEG5]: Is this ok in this form?

1.3.1 Application Information (Appinfo) (Level 1, Scorable)

Description:

Facilitates the running of interactive applications with additional administrative privileges. If this service is stopped, users will be unable to launch applications with the additional administrative privileges they may require to perform desired user tasks.

Rationale:

Disabling unnecessary services reduce the surface area of a system running SQL Server.

Remediation:

Open a command prompt or powershell window and issue the following command:

```
C:\>sc.exe stop Appinfo  
C:\>sc.exe config Appinfo start= disabled
```

Audit:

Open a powershell window and issue the following command:

```
PS C:\>Get-WmiObject -Class Win32_Service -Filter "Name='Appinfo'"
```

Verify that the following two lines are present in the output:

```
StartMode : Disabled  
State      : Stopped
```

Default Value:

This service is set to Manual by default.

References:

1.3.2 Application Layer Gateway (ALG) (Level 1, Scorable)

Description:

Provides support for 3rd party protocol plug-ins for Internet Connection Sharing

Rationale:

Disabling unnecessary services reduce the surface area of a system running SQL Server.

Remediation:

Open a command prompt or powershell window and issue the following command:

```
C:\>sc.exe stop ALG
C:\>sc.exe config ALG start= disabled
```

Audit:

Open a powershell window and issue the following command:

```
PS C:\>Get-WmiObject -Class Win32_Service -Filter "Name='ALG' "
```

Verify that the following two lines are present in the output:

```
StartMode : Disabled
State     : Stopped
```

Default Value:

This service is set to Manual by default.

References:

1.3.3 ASP.NET State Service (aspnet_state) (Level 1, Scorable)

Description:

Provides support for out-of-process session states for ASP.NET. If this service is stopped, out-of-process requests will not be processed. If this service is disabled, any services that explicitly depend on it will fail to start.

Rationale:

Disabling unnecessary services reduce the surface area of a system running SQL Server.

Remediation:

Open a command prompt or powershell window and issue the following command:

```
C:\>sc.exe stop aspnet_state
C:\>sc.exe config aspnet_state start= disabled
```

Audit:

Open a powershell window and issue the following command:

```
PS C:\>Get-WmiObject -Class Win32_Service -Filter "Name='aspnet_state' "
```

Verify that the following two lines are present in the output:

```
StartMode : Disabled
State     : Stopped
```

Default Value:

This service is set to Manual by default.

1.3.4 Computer Browser (browser) (Level 1, Scorable)

Description:

Maintains an updated list of computers on the network and supplies this list to computers designated as browsers. If this service is stopped, this list will not be updated or maintained. If this service is disabled, any services that explicitly depend on it will fail to start.

Rationale:

Disabling unnecessary services reduce the surface area of a system running SQL Server.

Remediation:

Open a command prompt or powershell window and issue the following command:

```
C:\>sc.exe stop browser
C:\>sc.exe config browser start= disabled
```

Audit:

Open a powershell window and issue the following command:

```
PS C:\>Get-WmiObject -Class Win32_Service -Filter "Name='browser'"
```

Verify that the following two lines are present in the output:

```
StartMode : Disabled
State      : Stopped
```

Default Value:

This service is set to Disabled by default.

1.3.5 Distributed Transaction Coordinator (MsDtc) (Level 1, Scorable)

Description:

Coordinates transactions that span multiple resource managers, such as databases, message queues, and file systems. If this service is stopped, these transactions will fail. If this service is disabled, any services that explicitly depend on it will fail to start. This service is required for distributed transactions.

Rationale:

Disabling unnecessary services reduce the surface area of a system running SQL Server.

Remediation:

Open a command prompt or powershell window and issue the following commands:

```
C:\>sc.exe stop msdtc
C:\>sc.exe config msdtc start= disabled
```

Audit:

Open a powershell window and issue the following command:

```
PS C:\>Get-WmiObject -Class Win32_Service -Filter "Name='msdtc'"
```

Verify that the following two lines are present in the output:

```
StartMode : Disabled
State      : Stopped
```

Default Value:

This service is set to Automatic/Delayed by default.

1.3.6 Function Discovery Provider Host (fdpHost) (Level 1, Scorable)

Description:

The FDPHOST service hosts the Function Discovery (FD) network discovery providers. These FD providers supply network discovery services for the Simple Services Discovery Protocol (SSDP) and Web Services – Discovery (WS-D) protocol. Stopping or disabling the FDPHOST service will disable network discovery for these protocols when using FD. When this service is unavailable, network services using FD and relying on these discovery protocols will be unable to find network devices or resources.

Rationale:

Disabling unnecessary services reduce the surface area of a system running SQL Server.

Remediation:

Open a command prompt or powershell window and issue the following command:

```
C:\>sc.exe stop fdpHost  
C:\>sc.exe config fdpHost start= disabled
```

Audit:

Open a powershell window and issue the following command:

```
PS C:\>Get-WmiObject -Class Win32_Service -Filter "Name='fdpHost' "
```

Verify that the following two lines are present in the output:

```
StartMode : Disabled  
State      : Stopped
```

Default Value:

This service is set to Manual by default.

1.3.7 Function Discovery Resource Publication (FDResPub) (Level 1, Scorable)

Description:

Publishes this computer and resources attached to this computer so they can be discovered over the network. If this service is stopped, network resources will no longer be published and they will not be discovered by other computers on the network.

Rationale:

Disabling unnecessary services reduce the surface area of a system running SQL Server.

Remediation:

Open a command prompt or powershell window and issue the following command:

```
C:\>sc.exe stop FDResPub  
C:\>sc.exe config FDResPub start= disabled
```


Audit:

Open a powershell window and issue the following command:

```
PS C:\>Get-WmiObject -Class Win32_Service -Filter "Name='FDResPub'"
```

Verify that the following two lines are present in the output:

```
StartMode : Disabled  
State      : Stopped
```

Default Value:

This service is set to Manual by default.

1.3.8 Internet Connection Sharing (ICS) (SharedAccess) (Level 1, Scorable)

Description:

Provides network address translation, addressing, name resolution and/or intrusion prevention services for a home or small office network.

Rationale:

Disabling unnecessary services reduce the surface area of a system running SQL Server.

Remediation:

Open a command prompt or powershell window and issue the following command:

```
C:\>sc.exe stop SharedAccess  
C:\>sc.exe config SharedAccess start= disabled
```

Audit:

Open a powershell window and issue the following command:

```
PS C:\>Get-WmiObject -Class Win32_Service -Filter "Name='SharedAccess'"
```

Verify that the following two lines are present in the output:

```
StartMode : Disabled  
State      : Stopped
```

Default Value:

This service is set to Disabled by default.

1.3.9 Link-Layer Topology Discovery Mapper (lltdsvc) (Level 1, Scorable)

Description:

Creates a Network Map, consisting of PC and device topology (connectivity) information, and metadata describing each PC and device. If this service is disabled, the Network Map will not function properly.

Rationale:

Disabling unnecessary services reduce the surface area of a system running SQL Server.

Remediation:

Open a command prompt or powershell window and issue the following command:

```
C:\>sc.exe stop lltdsvc
```

```
C:\>sc.exe config lltdsvc start= disabled
```

Audit:

Open a powershell window and issue the following command:

```
PS C:\>Get-WmiObject -Class Win32_Service -Filter "Name='lltdsvc'"
```

Verify that the following two lines are present in the output:

```
StartMode : Disabled
State      : Stopped
```

Default Value:

This service is set to Manual by default.

1.3.10 *Multimedia Class Scheduler (MMCSS) (Level 1, Scorable)*

Description:

Enables relative prioritization of work based on system-wide task priorities. This is intended mainly for multimedia applications. If this service is stopped, individual tasks resort to their default priority.

Rationale:

Disabling unnecessary services reduce the surface area of a system running SQL Server.

Remediation:

Open a command prompt or powershell window and issue the following command:

```
C:\>sc.exe stop MMCSS
C:\>sc.exe config MMCSS start= disabled
```

Audit:

Open a powershell window and issue the following command:

```
PS C:\>Get-WmiObject -Class Win32_Service -Filter "Name='MMCSS'"
```

Verify that the following two lines are present in the output:

```
StartMode : Disabled
State      : Stopped
```

Default Value:

This service is set to Manual by default.

1.3.11 *Print Spooler (Spooler) (Level 1, Scorable)*

Description:

Loads files to memory for later printing

Rationale:

Disabling unnecessary services reduce the surface area of a system running SQL Server.

Remediation:

Open a command prompt or powershell window and issue the following command:

```
C:\>sc.exe stop Spooler
C:\>sc.exe config Spooler start= disabled
```

Audit:

Open a powershell window and issue the following command:

```
PS C:\>Get-WmiObject -Class Win32_Service -Filter "Name='Spooler'"
```

Verify that the following two lines are present in the output:

```
StartMode : Disabled  
State      : Stopped
```

Default Value:

This service is set to Manual by default.

1.3.12 Remote Access Auto Connection Manager (RasAuto) (Level 1, Scorable)

Description:

Creates a connection to a remote network whenever a program references a remote DNS or NetBIOS name or address.

Rationale:

Disabling unnecessary services reduce the surface area of a system running SQL Server.

Remediation:

Open a command prompt or powershell window and issue the following command:

```
C:\>sc.exe stop RasAuto  
C:\>sc.exe config RasAuto start= disabled
```

Audit:

Open a powershell window and issue the following command:

```
PS C:\>Get-WmiObject -Class Win32_Service -Filter "Name='RasAuto'"
```

Verify that the following two lines are present in the output:

```
StartMode : Disabled  
State      : Stopped
```

Default Value:

This service is set to Manual by default.

1.3.13 Remote Access Connection Manager (RasMan) (Level 1, Scorable)

Description:

Manages dial-up and virtual private network (VPN) connections from this computer to the Internet or other remote networks. If this service is disabled, any services that explicitly depend on it will fail to start.

Rationale:

Disabling unnecessary services reduce the surface area of a system running SQL Server.

Remediation:

Open a command prompt or powershell window and issue the following command:

```
C:\>sc.exe stop RasMan  
C:\>sc.exe config RasMan start= disabled
```

Audit:

Open a powershell window and issue the following command:

```
PS C:\>Get-WmiObject -Class Win32_Service -Filter "Name='RasMan'"
```

Verify that the following two lines are present in the output:

```
StartMode : Disabled  
State      : Stopped
```

Default Value:

This service is set to Manual by default.

1.3.14 Remote Registry (RemoteRegistry) (Level 2, Scorable)

Description:

Enables remote users to modify registry settings on this computer. If this service is stopped, the registry can be modified only by users on this computer. If this service is disabled, any services that explicitly depend on it will fail to start. Microsoft Baseline Security Analyzer (MBSA) depends on this service to check installed hotfixes.

Rationale:

Disabling unnecessary services reduce the surface area of a system running SQL Server.

Remediation:

Open a command prompt or powershell window and issue the following command:

```
C:\>sc.exe stop RemoteRegistry  
C:\>sc.exe config RemoteRegistry start= disabled
```

Audit:

Open a powershell window and issue the following command:

```
PS C:\>Get-WmiObject -Class Win32_Service -Filter "Name='RemoteRegistry'"
```

Verify that the following two lines are present in the output:

```
StartMode : Disabled  
State      : Stopped
```

Default Value:

This service is set to Automatic by default.

1.3.15 Routing and Remote Access (RemoteAccess) (Level 1, Scorable)

Description:

Offers routing services to businesses in local area and wide area network environments.

Rationale:

Disabling unnecessary services reduce the surface area of a system running SQL Server.

Remediation:

Open a command prompt or powershell window and issue the following command:

```
C:\>sc.exe stop RemoteAccess  
C:\>sc.exe config RemoteAccess start= disabled
```

Audit:

Open a powershell window and issue the following command:

```
PS C:\>Get-WmiObject -Class Win32_Service -Filter "Name='RemoteAccess'"
```

Verify that the following two lines are present in the output:

```
StartMode : Disabled  
State      : Stopped
```

Default Value:

This service is set to Disabled by default.

1.3.16 *Telephony (TapiSrv) (Level 1, Scorable)*

Description:

Provides Telephony API (TAPI) support for programs that control telephony devices on the local computer and, through the LAN, on servers that are also running the service.

Rationale:

Disabling unnecessary services reduce the surface area of a system running SQL Server.

Remediation:

Open a command prompt or powershell window and issue the following command:

```
C:\>sc.exe stop TapiSrv  
C:\>sc.exe config TapiSrv start= disabled
```

Audit:

Open a powershell window and issue the following command:

```
PS C:\>Get-WmiObject -Class Win32_Service -Filter "Name='TapiSrv'"
```

Verify that the following two lines are present in the output:

```
StartMode : Disabled  
State      : Stopped
```

Default Value:

This service is set to Manual by default.

1.3.17 *Windows Audio (AudioSrv) (Level 1, Scorable)*

Description:

Manages audio for Windows-based programs. If this service is stopped, audio devices and effects will not function properly. If this service is disabled, any services that explicitly depend on it will fail to start.

Rationale:

Disabling unnecessary services reduce the surface area of a system running SQL Server.

Remediation:

Open a command prompt or powershell window and issue the following command:

```
C:\>sc.exe stop AudioSrv  
C:\>sc.exe config AudioSrv start= disabled
```

Audit:

Open a powershell window and issue the following command:

```
PS C:\>Get-WmiObject -Class Win32_Service -Filter "Name='AudioSrv'"
```

Verify that the following two lines are present in the output:

```
StartMode : Disabled  
State      : Stopped
```

Default Value:

This service is set to Manual by default.

1.3.18 Windows Audio Endpoint Builder (AudioEndpointBuilder) (Level 1, Scorable)

Description:

Manages audio devices for the Windows Audio service. If this service is stopped, audio devices and effects will not function properly. If this service is disabled, any services that explicitly depend on it will fail to start

Rationale:

Disabling unnecessary services reduce the surface area of a system running SQL Server.

Remediation:

Open a command prompt or powershell window and issue the following command:

```
C:\>sc.exe stop AudioEndpointBuilder  
C:\>sc.exe config AudioEndpointBuilder start= disabled
```

Audit:

Open a powershell window and issue the following command:

```
PS C:\>Get-WmiObject -Class Win32_Service -Filter  
"Name='AudioEndpointBuilder'"
```

Verify that the following two lines are present in the output:

```
StartMode : Disabled  
State      : Stopped
```

Default Value:

This service is set to Manual by default.

1.3.19 Windows Color System (WcsPlugInService) (Level 1, Scorable)

Description:

The WcsPlugInService service hosts third-party Windows Color System color device model and gamut map model plug-in modules. These plug-in modules are vendor-specific extensions to the Windows Color System baseline color device and gamut map models. Stopping or disabling the WcsPlugInService service will disable this extensibility feature, and the Windows Color System will use its baseline model processing rather than the vendor's desired processing. This might result in inaccurate color rendering.

Rationale:

Disabling unnecessary services reduce the surface area of a system running SQL Server.

Remediation:

Open a command prompt or powershell window and issue the following command:

```
C:\>sc.exe stop WcsPlugInService
C:\>sc.exe config WcsPlugInService start= disabled
```

Audit:

Open a powershell window and issue the following command:

```
PS C:\>Get-WmiObject -Class Win32_Service -Filter "Name='WcsPlugInService'"
```

Verify that the following two lines are present in the output:

```
StartMode : Disabled
State      : Stopped
```

Default Value:

This service is set to Manual by default.

1.3.20 Windows Font Cache Service (FontCache) (Level 1, Scorable)

Description:

Optimizes performance of applications by caching commonly used font data. Applications will start this service if it is not already running. It can be disabled, though doing so will degrade application performance.

Rationale:

Disabling unnecessary services reduce the surface area of a system running SQL Server.

Remediation:

Open a command prompt or powershell window and issue the following command:

```
C:\>sc.exe stop FontCache
C:\>sc.exe config FontCache start= disabled
```

Audit:

Open a powershell window and issue the following command:

```
PS C:\>Get-WmiObject -Class Win32_Service -Filter "Name='FontCache'"
```

Verify that the following two lines are present in the output:

```
StartMode : Disabled
State      : Stopped
```

Default Value:

This service is set to Manual by default.

1.3.21 Windows Remote Management (WS-Management) (WinRM) (Level 2, Scorable)

Description:

Windows Remote Management (WinRM) service implements the WS-Management protocol for remote management. WS-Management is a standard web services protocol used for remote software and hardware management. The WinRM service listens on the network for WS-Management requests and processes them. The WinRM Service needs to be configured with a listener using winrm.cmd command line tool or through Group Policy in order for it to listen over the network. The WinRM service provides access to WMI data

and enables event collection. Event collection and subscription to events require that the service is running. WinRM messages use HTTP and HTTPS as transports. The WinRM service does not depend on IIS but is preconfigured to share a port with IIS on the same machine. The WinRM service reserves the /wsman URL prefix. To prevent conflicts with IIS, administrators should ensure that any websites hosted on IIS do not use the /wsman URL prefix.

This service is required by PowerShell v2 Remoting.

Rationale:

Disabling unnecessary services reduce the surface area of a system running SQL Server.

Remediation:

Open a command prompt or powershell window and issue the following command:

```
C:\>sc.exe stop WinRM
C:\>sc.exe config WinRM start= disabled
```

Audit:

Open a powershell window and issue the following command:

```
PS C:\>Get-WmiObject -Class Win32_Service -Filter "Name='WinRM'"
```

Verify that the following two lines are present in the output:

```
StartMode : Disabled
State      : Stopped
```

Default Value:

This service is set to Automatic by default.

1.3.22 WinHTTP Web Proxy Auto-Discovery Service (WinHttpAutoProxySvc) (Level 1, Scorable)

Description:

WinHTTP implements the client HTTP stack and provides developers with a Win32 API and COM Automation component for sending HTTP requests and receiving responses. In addition, WinHTTP provides support for auto-discovering a proxy configuration via its implementation of the Web Proxy Auto-Discovery (WPAD) protocol.

Rationale:

Disabling unnecessary services reduce the surface area of a system running SQL Server.

Remediation:

Open a command prompt or powershell window and issue the following command:

```
C:\>sc.exe stop WinHttpAutoProxySvc
C:\>sc.exe config WinHttpAutoProxySvc start= disabled
```

Audit:

Open a powershell window and issue the following command:

```
PS C:\>Get-WmiObject -Class Win32_Service -Filter
"Name='WinHttpAutoProxySvc'"
```

Verify that the following two lines are present in the output:

StartMode : Disabled
State : Stopped

Default Value:

This service is set to Manual by default.

1.3.23 Distributed Link Tracking Client (TrkWks) (Level 1, Scorable)

Description:

Maintains links between NTFS files within a computer or across computers in a network.

Rationale:

Disabling unnecessary services reduce the surface area of a system running SQL Server.

Remediation:

Open a command prompt or powershell window and issue the following command:

```
C:\>sc.exe stop TrkWks  
C:\>sc.exe config TrkWks start= disabled
```

Audit:

Open a powershell window and issue the following command:

```
PS C:\>Get-WmiObject -Class Win32_Service -Filter "Name='TrkWks' "
```

Verify that the following two lines are present in the output:

StartMode : Disabled
State : Stopped

Default Value:

This service is set to Manual by default.

1.3.24 Windows Installer (msiserver) (Level 1, Scorable)

Description:

Adds, modifies, and removes applications provided as a Windows Installer (*.msi) package. If this service is disabled, any services that explicitly depend on it will fail to start.

Enable the Windows Installer service only for the duration of planned software installation.

Rationale:

Disabling unnecessary services reduce the surface area of a system running SQL Server.

Remediation:

Open a command prompt or powershell window and issue the following command:

```
C:\>sc.exe stop msiserver  
C:\>sc.exe config msiserver start= disabled
```

Audit:

Open a powershell window and issue the following command:

```
PS C:\>Get-WmiObject -Class Win32_Service -Filter "Name='msiserver' "
```

Verify that the following two lines are present in the output:

```
StartMode : Disabled  
State     : Stopped
```

Default Value:

This service is set to Manual by default.

1.4 Service Account Management

1.4.1 Dedicated SQL Server Service Account (Level 1, Not Scorable)

Description:

Do not use the SQL Server service accounts for any other applications.

Rationale:

Using a SQL Server service account for other purposes as well defeats the principle of least privilege and poses a threat for both the SQL Server instance and the other application.

Remediation:

Create separate service account

Audit:

Verify if the SQL Server service account is not used for other purposes.

Default Value:

Not Applicable

References:

1.4.2 Local users group membership (Level 1, Scorable)

Description:

If you use local user accounts to run SQL Server services on the machine, they should only belong to the Users group, and not be designated as an administrator.

Rationale:

Take this measure to ensure that local service account is only a user on the machine and not an administrator.

Remediation:

If the local service account is an administrator on the machine, consider removing administrator privileges. Perform the steps outlined in the Audit section below to view and alter these privileges as necessary.

Audit:

Perform the following actions on the server to identify SQL Server port assignments:

1. Run compmgmt.msc from the Run menu.
2. Expand Local Users and Groups.

3. Expand Groups.
4. Expand the Administrators group to view its properties. Ensure that the accounts that run local services are not assigned to the Administrators group.

Default Value:

By default, user accounts that run services are granted the 'Log On as a Service' right, and do not require administrative permissions.

References:

<http://msdn.microsoft.com/en-us/library/ms143504.aspx>

1.4.3 Domain service account group membership (Level 1, Not Scorable)

Description:

If you use domain user accounts to run SQL Server services on the machine, they should only belong to non-privileged domain groups.

Rationale:

SQL Server service accounts do not typically require elevated domain privileges. Ensuring that SQL Server service account and the SQL Agent account run under the lowest privileges necessary lowers the risk to the network if these accounts are compromised.

Remediation:

Verify the domain permissions if either the SQL Server service or SQL Server agent account runs under a domain user. If the account has elevated permissions, consider revising the security scheme to encompass only the necessary permissions.

Audit:

The domain user permissions may be viewed through Active Directory or run the following command in a command prompt:

```
C:\>net user _serviceaccount /domain
```

Default Value:

By default, the SQL Server service and SQL Server agent do not run under domain accounts.

References:

<http://msdn.microsoft.com/en-us/library/ms191543.aspx>

1.4.4 Service Account Dedicated to the SQL Server Instance (Level 2, Not Scorable)

Description:

Use the SQL Server service accounts exclusively for a particular SQL Server instance. Do not share service accounts across SQL Server instances.

Rationale:

Comment [BEG6]: FOR CONSENSUS: We can have a debate here if we want to take the hardcore approach and say that level 2 config means you don't share an account even across SQL Server instances. Although it's fine for security, enterprises with 1000+ SQL Server instances might want to kill me.

Using a SQL Server service account dedicated to a particular instance supports the principle of least privilege and reduces the impact of brute-force attacks which result in service account lockout.

Remediation:

Create separate service accounts for every SQL Server instance.

Audit:

Create a list of SQL Server services and accounts running them. Ensure there's no account running more than one service.

Default Value:

Not Applicable

References:

1.4.5 Different Service Accounts (Level 2, Scorable)

Description:

Use different service accounts for running different SQL Server services (SQL Server, SQL Server Agent, etc).

Rationale:

Using different accounts supports the principle of least privilege and minimizes attack surface.

Remediation:

Configure SQL Server Agent to use different service account:

1. Launch SQL Server Configuration Manager.
2. Go to SQL Server Services or SQL Server Services(32bit).
3. Double-click on SQL Server Agent (INSTANCENAME).
4. On the Log On tab, select This account and set a different account than the one running the SQL Server (INSTANCENAME) service.
5. Click Apply.
6. Click Restart.
7. Click OK.

Audit:

1. Launch SQL Server Configuration Manager.
2. Go to SQL Server Services or SQL Server Services(32bit).
3. Verify that the services listed are not sharing a domain account.

Default Value:

This property is blank during installation, it provides the option to specify separate service account and also to specify the same account for all services.

Comment [BEG7]: FOR CONSENSUS:
Yet another point to debate. Pro: security; Con: in most of the cases, minimal benefit compared to the overhead.

References:

1.5 SQL Server Service Account Rights

1.5.1 MSSQL Server Service Account (Level 1, Not Scorable)

Description:

Use a low-privileged Local or Domain account for the MS SQL Server service.

Rationale:

The services account should only be a domain account if the SQL Server requires remote communications with other domain systems such as those used for backup over the network. Otherwise, a local user account should be used.

Remediation:

If network access is not necessary change the account that runs the SQL Server Service to a local user account. If network access is required, use a domain account with permissions only necessary to access required network resources.

Audit:

Perform the following to determine if the system is configured as recommended:

1. Run `services.msc` from the Run menu.
2. Right click the SQL Server service and select Properties.
3. Navigate to the Log On tab. This tab identifies the account the service runs under.

Validate the group membership of the account running the SQL Server service.

Default Value:

By default, a non-privileged service account is offered by the installer.

1.5.2 Grant the SQL Server Service Account Only Necessary Privileges (Level 1, Scorable)

Description:

In order to operate properly, the SQL Server service account must be given certain operating system level permissions:

Log on as a service (SeServiceLogonRight)

Replace a process-level token (SeAssignPrimaryTokenPrivilege)

Bypass traverse checking (SeChangeNotifyPrivilege)

Adjust memory quotas for a process (SeIncreaseQuotaPrivilege)

Regarding other additional privileges needed, refer to Recommendations 1.5.3 and 1.5.4 . No other privileges should be granted to the service account.

Rationale:

SQL Server requires exactly the privileges listed above to work properly, no need for additional privileges.

Remediation:

Remove any additional privileges granted to the service account.

1. execute gpedit.msc.
2. Navigate to the following path in the Group Policy editor: Computer Configuration\Windows Settings\Security Settings\Local Policies\User Rights Assignment\.
3. Remove any additional privileges granted to the service account.

Audit:

Perform the following to determine if the system is configured as recommended:

1. execute gpedit.msc.
2. Navigate to the following path in the Group Policy editor: Computer Configuration\Windows Settings\Security Settings\Local Policies\User Rights Assignment\.
3. Verify that only privileges listed here are granted to the service account.

Default Value:

By default, no additional permissions are granted to the service account.

References:

<http://msdn.microsoft.com/en-us/library/ms143504.aspx>

1.5.3 Lock Pages in Memory Privilege (Level 1, Scorable)

Description:

SQL Server Standard Edition version 10.00.2714 (SP1 CU2) or later or Enterprise Edition (all versions) is able to prevent the service from being paged out from memory if this privilege is granted.

Rationale:

SQL Server runs more stable and reliable if granted the Lock Pages in Memory (SeLockMemoryPrivilege). Always use this option with maximum server memory limit to avoid interference with other applications and the operating system.

Remediation:

If you use Standard Edition, add trace flag 845 to the startup options.

Open SQL Server Configuration Manager, select the SQL Server service and open up its properties. To the startup options add ;-T845, without spaces.

Grant the lock pages in memory privilege to the SQL Server service account:

1. execute gpedit.msc.
2. Navigate to the following path in the Group Policy editor: Computer Configuration\Windows Settings\Security Settings\Local Policies\User Rights Assignment\.
3. Double-click on the privilege Lock Pages in Memory.

Comment [BEG8]: NEW

4. Add the SQL Server service account to it.
5. Click OK.
6. Run gpupdate /force.
7. Restart the SQL Server service after you verified that the privileges got into effect.

Audit:

Open the ERRORLOG file being written by SQL Server. Check if it contains a line at the startup part "Using locked pages for buffer pool".

Default Value:

This option is not set by default.

References:

1.5.4 *Perform Volume Maintenance Tasks Privilege* (Level 2, Scorable)

Comment [BEG9]: NEW. Topic for consensus...

Description:

SQL Server Enterprise Edition is able to allocate disk space for database files instantly if this privilege is granted, by bypassing the zeroing out of the recently allocated space on disk. This is a performance boost and a possible security threat.

Rationale:

This disk allocation method may allow someone with dbo privileges on any database to allocate disk space previously assigned to another object inside or outside SQL Server and retrieve data stored by it.

Remediation:

1. execute gpedit.msc.
2. Navigate to the following path in the Group Policy editor: Computer Configuration\Windows Settings\Security Settings\Local Policies\User Rights Assignment\.
3. Double-click on the privilege Perform Volume Maintenance Tasks.
4. Remove the SQL Server service account from it.
5. Click OK.
6. Run gpupdate /force.
7. Restart the SQL Server service after you verified that the privileges revocation got into effect.

Audit:

1. execute gpedit.msc.
2. Navigate to the following path in the Group Policy editor: Computer Configuration\Windows Settings\Security Settings\Local Policies\User Rights Assignment\.
3. Check that the privilege Perform Volume Maintenance Tasks. Is not granted to the SQL Server service account.

Default Value:

References:

<http://blogs.msdn.com/b/askjay/archive/2011/03/11/why-is-using-instant-file-initialization-on-my-sql-server-a-potential-security-issue.aspx>

1.5.5 Password Management (Level 1, Scorable)

Description:

Check the Password never expires option for the SQL Server Service service account and establish a process to change the password manually on a regular basis. Use a complex password, at least 15 characters long. Applicable only if the SQL Server service account is a domain or local user account.

Comment [BEG10]: Shall I give into so specific recommendation or just say 'a long one'?

Rationale:

Changing the password on a regular basis reduces the risk of password compromise. Disabling the automatic password expiration prevents an accidental service downtime.

Remediation:

If it's a domain account:

1. Open the Active Directory Users and Computers management snap-in by running dsa.msc. (This snap-in is not available by default on non domain controller servers; you may need to install it.)
2. Locate the service account using the find function.
3. Open the service account by double-clicking on it and go to the Account tab.
4. Check the Password never expires option.

If it's a local account:

1. Open the Computer Management management snap-in by running compmgmt.msc.
2. Go to the Local Users and Computers / Users node.
3. Double-click on the service account in the right pane.
4. Go to General tab and check the Password never expires option.

Audit:

Open a powershell window and enter the following command in case of a local account:

```
C:\>net user <serviceaccount>
```

Open a powershell window and enter the following command in case of a domain account:

```
C:\>net user <serviceaccount> /domain
```

In the output, check if the Password expires Never line appears. Check if the Password last set line contains a date later than current date minus the maximum password age defined in the password policy.

Default Value:

By default, the Password never expires option is not set for an account.

References:

1.6 SQL Server Agent Service Account Rights

1.6.1 SQL Server Agent Service Account (Level 1, Scorable)

Description:

Use a low-privileged domain account for SQL Server Agent if replication or other inter-server connection is required.

Rationale:

Replication and other inter-server communications require the SQL Server Agent service account to be a domain account.

Remediation:

Ensure that the SQL Agent uses a low-privilege domain account.

Audit:

Perform the following to determine if the system is configured as recommended:

1. Run `services.msc` from the Run menu.
2. Right click the SQL Server service and select Properties.
3. Navigate to the Log On tab. This tab identifies the account the service runs under.

Default Value:

By default the SQL Agent runs under a local system account.

References:

<http://msdn.microsoft.com/en-us/library/ms191543.aspx>

1.6.2 Grant the SQL Server Agent Account Necessary Privileges (Level 1, Scorable)

Description:

In order to operate properly, the SQL Server service account must be given certain operating system level permissions:

Log on as a service (SeServiceLogonRight)

Replace a process-level token (SeAssignPrimaryTokenPrivilege)

Bypass traverse checking (SeChangeNotifyPrivilege)

Adjust memory quotas for a process (SeIncreaseQuotaPrivilege)

Regarding other additional privileges needed, refer to Recommendations 1.5.3 and 1.5.4 . No other privileges should be granted to the service account.

Rationale:

SQL Server Agent requires exactly the privileges listed above to work properly, no need for additional privileges.

Remediation:

Remove any additional privileges granted to the service account.

1. execute gpedit.msc.
2. Navigate to the following path in the Group Policy editor: Computer Configuration\Windows Settings\Security Settings\Local Policies\User Rights Assignment\.
3. Remove any additional privileges granted to the service account.

Audit:

Perform the following to determine if the system is configured as recommended:

1. execute gpedit.msc.
2. Navigate to the following path in the Group Policy editor: Computer Configuration\Windows Settings\Security Settings\Local Policies\User Rights Assignment\.
3. Verify that only privileges listed here are granted to the service account.

Default Value:

By default, no additional permissions are granted to the service account.

References:

<http://msdn.microsoft.com/en-us/library/ms143504.aspx>

1.6.3 SQL Server Agent Proxy Accounts (Level 1, Not Scorable)

Description:

A SQL Server Agent proxy allows SQL Server Agent job steps to run under the security context of a Windows user other than the service account.

Consider using proxy accounts if there are only a limited number of SQL Server Agent job steps requiring certain elevated permissions. Limit access to proxies only to the necessary logins. Do not grant access to CmdEXECUTE, ActiveX and PowerShell subsystems or proxies unless it is needed.

Rationale:

Using proxy accounts instead of granting excessive permissions to the SQL Server Agent service account prevents job steps not requiring special permissions from performing actions they are not allowed to do.

Remediation:

Create a new Windows account and grant it the necessary special permissions. Create new proxy for it by using the sp_add_proxy stored procedure and grant access to the new proxy with the sp_grant_login_to_proxy SP.

Revoke excessive permissions from the SQL Server Agent service account.

Audit:

Create a list of permissions granted to SQL Server Agent service account. Check if there's any special privilege and if that is used only for a specific subset of the agent job steps.

Default Value:

By default the SQL Server agent proxy is not given elevated permissions.

References:

<http://msdn.microsoft.com/en-us/library/ms189064.aspx>

1.7 Integration Services service account rights

1.7.1 SQL Server Integration Services Service Account (Level 1, Scorable)

Description:

Use a low-privileged account for SQL Server Integration Services like the built-in Network Service or a non-privileged domain user account.

Rationale:

Using a low-privileged account supports the principle of least privilege.

Remediation:

Ensure that the SQL Server Integration Services service uses a low-privilege domain account.

Audit:

Perform the following to determine if the system is configured as recommended:

4. Run `services.msc` from the Run menu.
5. Right click the SQL Server Integration Services service and select Properties.

Navigate to the Log On tab. This tab identifies the account the service runs under.

Default Value:

NT_AUTHORITY\NetworkService

References:

1.7.2 Grant the SQL Server Integration Services Service Account Only Necessary Privileges (Level 1, Scorable)

Description:

In order to operate properly, the SQL Server Integration Services service account must be given certain operating system level permissions:

Log on as a service (SeServiceLogonRight)

No other privileges should be granted to the service account.

Rationale:

SQL Server Integration Services requires exactly the privileges listed above to work properly, no need for additional privileges.

Remediation:

Remove any additional privileges granted to the service account.

4. execute gpedit.msc.
5. Navigate to the following path in the Group Policy editor: Computer Configuration\Windows Settings\Security Settings\Local Policies\User Rights Assignment\.
6. Remove any additional privileges granted to the service account.

Audit:

Perform the following to determine if the system is configured as recommended:

4. execute gpedit.msc.
5. Navigate to the following path in the Group Policy editor: Computer Configuration\Windows Settings\Security Settings\Local Policies\User Rights Assignment\.
6. Verify that only privileges listed here are granted to the service account.

Default Value:

By default, no additional permissions are granted to the service account.

References:

<http://msdn.microsoft.com/en-us/library/ms143504.aspx>

1.8 SQL Server Installation and Patches

1.8.1 Domain Controller Installation (Level1, Scorable)

Description:

Avoid installing SQL Server on a domain controller.

Rationale:

If SQL Server were installed on a domain controller, a successful attack against the database could potentially compromise all domain resources.

Remediation:

Install SQL Server onto a new server and move databases. Uninstall SQL Server services from the domain controller.

Audit:

Perform an audit on the domain controller to determine if the SQL Server instances are installed. If they are installed but not being used, uninstall them. If SQL Server is being used on the domain controller, develop a plan to migrate the instances to new servers.

Default Value:

By default, SQL Server is able to be installed on a domain controller, but the installation process includes a warning indicating that the instance is being installed on a DC.

References:

<http://msdn.microsoft.com/en-us/library/ms143719%28SQL.90%29.aspx>

1.8.2 *Service Packs and Hotfixes (Level 1, Not Scorable)*

Description:

SQL Server patches and hot fixes contain program updates that fix found issues in the software. These come in three types: service packs, cumulative updates and hotfixes. Establish a software update process for SQL Server, determining the maximum latency for Service Pack installations and the rollout policy for cumulative updates and hotfixes.

Rationale:

Patches and hotfixes typically contain specific fixes for security vulnerabilities.

Remediation:

Ensure the current SQL Server service pack and hotfixes or cumulative updates are installed. Make sure to test these fixes in your test environments before updating production instances. Visit <http://support.microsoft.com/sp> to find and download the latest SQL Server service packs.

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.

Default Value:

Service packs and patches are not installed by default.

References:

<http://support.microsoft.com/sp>

<http://support.microsoft.com/kb/968382>

1.8.3 *SQL Server Agent Service Startup (Level 1, Scorable)*

Description:

Set the SQL Server Agent service to automatic start.

Rationale:

The SQL Server Agent service provides a widely used scheduler for SQL Server. Without it running, audit traces other than default trace and database backups won't start.

Remediation:

Comment [BEG11]: Shall we add that MSFT officially recommends the installation of SPs and only those hotfixes which affects your installation, that is, not to install all hotfixes and CUs?

During the installation, set the SQL Server Agent service to Automatic startup type.

OR

Open a command prompt or powershell window and run the following command in case of a default instance:

```
C:\>sc.exe config SQLSERVERAGENT start= auto
```

and the following command in case of a named instance:

```
C:\>sc.exe config SQLAGENT$INSTANCENAME start= auto
```

Audit:

Open a powershell window and issue the following command:

```
PS C:\>Get-WmiObject -Class Win32_Service -Filter "Name='SQLSERVERAGENT' "
```

Verify that the following line is present in the output:

```
StartMode : Automatic
```

Default Value:

References:

1.8.4 Instance Naming Conventions (Level 1, Not Scorable)

Description:

In naming SQL Server instances, limit the instance name to less than 16 characters with no reference to a version number or purpose or other sensitive information.

Rationale:

Version or other sensitive information in the server name makes it easier for an attacker to develop an attack strategy against the server.

Remediation:

Reinstall your SQL Server instances if the name indicates the purposes of the machine. SQL Server instance name cannot be changed after installation.

Audit:

Review the names and naming conventions of the public-facing production SQL Servers on the network.

Default Value:

SQL Server default instance name is MSSQLSERVER.

References:

1.8.5 Inventory of SQL Server Instances (Level 1, Not Scorable)

Description:

Keep an inventory of all versions, editions and languages of SQL Server, including date of version upgrades. A CMDB covering this information is also a solution.

Rationale:

Keeping an active inventory of all SQL Server instances on your network is a good first step to ensuring that you've secured your SQL Server environment. This list should be kept up-to-date. There are some good third party applications that you can use to inventory your SQL Server environment.

Remediation:

Create an inventory of SQL Server instances if one has not already been created.

Audit:

Use any of the above listed tools to create an inventory of the SQL Server instances on the network.

Default Value:

By default, no product ships with SQL Server to inventory installed instances.

1.8.6 Sample databases (Level 1, Scorable)

Description:

Remove any sample databases if they are installed on the SQL Server instance.

Rationale:

Sample databases have well known security models. Leaving these databases installed and not fully securing them could lead to security risks. Also, these databases shouldn't be installed in production environments as they will not be used.

Remediation:

If sample databases are installed, take backups if necessary and drop the databases.

Audit:

Run the following code snippet to determine if any sample databases are installed.

```
SELECT name
FROM sys.databases
WHERE
    name LIKE 'AdventureWorks%' OR
    name = 'pubs' OR
    name = 'Northwind';
```

Default Value:

Sample databases are not installed by default.

References:

1.8.7 Lock Down sa Account (Level 1, Scorable)

Description:

The `sa` account is a widely known and often widely used SQL Server account with `sysadmin` privileges. This account should be renamed to something that is not easily identifiable as the `sa` account and disabled. Avoid using the `sa` account for any administrative purposes. Create individual administrative accounts instead.

Rationale:

It is more difficult to script attacks against the `sa` account if the username is not known and/or disabled.

Remediation:

Use the following code snippet to disable the `sa` login and rename it.

```
ALTER LOGIN sa DISABLE;  
ALTER LOGIN sa WITH NAME = different_user;
```

Audit:

Use the following code snippet to determine if the `sa` account is renamed and disabled.

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

Default Value:

By default the `sa` account is not enabled. The `sa` account is only available for use under the mixed-mode authentication scheme.

References:

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

1.8.8 Strong sysadmin password (Level 1, Not Scorable)

Description:

Use a strong password for the `sa` login account and all logins in the `sysadmin` server role.

Rationale:

A strong password for the `sa` login account is required regardless of which mode is chosen and regardless of whether the `sa` account is disabled. Strong passwords provide higher level of security against attacks.

Remediation:

Design a policy for complexity for administrative passwords. Evaluate all administrative passwords to ensure they conform to the complexity policy.

Audit:

Perform an audit of the `sa` password of all network SQL Server instances. Change any passwords that do not meet your complexity requirements.

Default Value:

There is no default value for passwords.

References:

1.8.9 SQL Server Program Directory Permissions (Level 1, Scorable)

Description:

Modify the permissions to the `%ProgramFiles%\Microsoft SQL Server\<InstanceId>\Binn` directory. Assign the SQL Server service account Read+execute permission. Remove the Users group's permission.

Rationale:

Securing the directory containing the binaries for SQL Server prevents any tampering with the executables.

Remediation:

1. Browse to the directory in Windows Explorer and open the Properties dialog.
2. Go to Security tab and set the NTFS settings.

Audit:

1. Browse to the directory in Windows Explorer and open the Properties dialog.
2. Go to Security tab and review the NTFS settings.

Default Value:

By default, the administrators and users groups have permissions to this folder.

References:

<http://msdn.microsoft.com/en-us/library/ms143547.aspx>
[http://msdn.microsoft.com/en-us/library/ms143504%28v=SQL.100%29.aspx#Reviewing ACLs](http://msdn.microsoft.com/en-us/library/ms143504%28v=SQL.100%29.aspx#Reviewing_ACLs)

1.8.10 SQL Server database instance directory permissions (Level 1, Scorable)

Description:

Delete or secure old setup files. Protect files in the `%ProgramFiles%\Microsoft SQL Server\MSSQL10.<INSTANCENAME>\MSSQL\Install`, e.g., `sqlstp.log`, `sqlsp.log` and `setup.iss`.

Rationale:

If the current system was upgraded from SQL Server version 2000, check `setup.iss` in the `%SystemRoot%` folder and the `sqlstp.log` in the Windows Temp folder for passwords.

Remediation:

Secure the permissions on this folder or remove sensitive files to disallow any potential information disclosure.

Audit:

Navigate to the `%ProgramFiles%\Microsoft SQL Server\MSSQL10.<INSTANCENAME%\MSSQL\Install` directory to review the folder permissions.

Default Value:

By default, the Administrators and Users local groups have permissions to this folder.

References:

<http://msdn.microsoft.com/en-us/library/ms143547.aspx>

1.8.11 SQL Server Data Directory Permissions (Level 1, Scorable)

Description:

Modify the permissions to the directory/directories storing the SQL Server database files. Assign the SQL Server service account Full Control permission. Remove the Users group's permission.

Rationale:

Securing the directory containing the databases for SQL Server prevents unauthorized access to data on the file system level.

Remediation:

1. Determine the directories in stake by checking the default data directory and the location of already existing data files:

```
SELECT db_name(database_id), physical_name AS filepath
FROM sys.master_files;
```

2. Browse to the directory in Windows Explorer and open the Properties dialog.
3. Go to Security tab and set the NTFS settings.

Audit:

1. Determine the directories in stake by checking the default data directory and the location of already existing data files:

```
SELECT db_name(database_id), physical_name AS filepath
FROM sys.master_files;
```

2. Browse to the directory in Windows Explorer and open the Properties dialog.
3. Go to Security tab and review the NTFS settings.

Default Value:

By default, the SQL Server account, the Administrators and Users local groups have permissions to this folder.

References:

<http://msdn.microsoft.com/en-us/library/ms143547.aspx>
http://msdn.microsoft.com/en-us/library/ms143504%28v=SQL.100%29.aspx#Reviewing_ACLs

1.8.12 SQL Server Log Directory Permissions (Level 1, Scorable)

Description:

Modify the permissions to the directory/directories storing the SQL Server errorlog and SQL Agent log files. Assign the SQL Server service account Full Control permission. Remove the Users group's permission.

Rationale:

Securing the directory containing the log files for SQL Server prevents potential information disclosure regarding SQL Server configuration and possible vulnerabilities.

Remediation:

1. Determine the SQL Server errorlog path by opening SQL Server Configuration Manager, selecting the SQL Server service and opening up its properties. Observe the value of startup parameters: the one starting with -e contains the location of the errorlog file.
2. Determine the SQL Agent log path in SSMS by browsing to the SQL Server Agent / Error Log node, right-clicking on it and choosing Configure. The location of the errorlog can be seen in the dialog box opened.
3. Browse to the directories in Windows Explorer and open the Properties dialog.
4. Go to Security tab and set the NTFS settings.

Audit:

1. Determine the SQL Server errorlog path by opening SQL Server Configuration Manager, selecting the SQL Server service and opening up its properties. Observe the value of startup parameters: the one starting with -e contains the location of the errorlog file.
2. Determine the SQL Agent log path in SSMS by browsing to the SQL Server Agent / Error Log node, right-clicking on it and choosing Configure. The location of the errorlog can be seen in the dialog box opened.
3. Browse to the directories in Windows Explorer and open the Properties dialog.
4. Go to Security tab and review the NTFS settings.

Default Value:

By default, the SQL Server account, the Administrators and Users local groups have permissions to this folder.

References:

<http://msdn.microsoft.com/en-us/library/ms143547.aspx>
http://msdn.microsoft.com/en-us/library/ms143504%28v=SQL.100%29.aspx#Reviewing_ACLs

1.8.13 *Monitor SQL Server (Level 1, Scorable)*

Description:

Set up monitoring for SQL Server.

Rationale:

Notifications about excessive use of resources or high rate of errors may prevent service interruptions and/or discover malicious activities.

Remediation:

Set up monitoring and define monitor thresholds based on a baseline measurement and available capacity.

Audit:

Verify if monitoring works as expected.

Default Value:

By default, no monitoring is set up for SQL Server.

References:

1.9 SQL Server Interface Configuration

1.9.1 *Disable Unnecessary Protocols (Level 1, Scorable)*

Description:

Configure the SQL Server service to listen only protocols used by clients. Keep the number of protocols used on the minimum.

Rationale:

Using fewer protocols eases the administration of SQL Server and minimizes the attack surface.

Remediation:

Open SQL Server Configuration Manager; go to the SQL Server Network Configuration. Ensure that only required protocols are enabled. Disable protocols not necessary.

Audit:

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

Ensure that only required protocols are enabled.

Default Value:

By default, SQL Server 2008 uses TCP/IP and Shared Memory protocols.

References:

1.9.2 SQL Server Ports (Level 1, Scorable)

Description:

Consider changing SQL Server default port from 1433.

Rationale:

Using a non-default port helps protect the database from attacks directed to the default port. SQL Server uses port 1433 for TCP traffic. Note that doing so will make the DAC (Default Administrator Connection) listen on a random port by default. Also, it might make benign applications, such as application firewalls, requiring special configuration.

Remediation:

If your SQL Server instance is listening on port 1433, consider changing it to a higher port that is not in use. Doing so can mask SQL Server from various vulnerability tools. Avoid using port 2433 as that was the traditional “hidden port” in SQL 2000.

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.

Default Value:

By default, SQL Server listens on to TCP/IP traffic on TCP port 1433.

References:

<http://msdn.microsoft.com/en-us/library/ms177440%28v=SQL.100%29.aspx>
<http://support.microsoft.com/kb/308091>

1.9.3 Limit Network Interfaces (Level 1, Scorable)

Description:

Limit SQL Server to listen only on those network interfaces where it is necessary.

Rationale:

Limiting interfaces used limits exposure of SQL Server.

Remediation:

If your SQL Server instance is listening on multiple interfaces, determine which ones are used by clients and configure SQL Server to not listen on any unused interface.

To control which network interfaces are used, open SQL Server Configuration Manager, go to SQL Server network Protocols, open TCP/IP and go to IP Addresses. Set Enabled to false for IP addresses not actually in use.

Audit:

Open a powershell window and run the following command:

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

Replace <portnum> with the actual port number SQL Server is listening on. Verify if only necessary IP addresses are in use.

Default Value:

By default, SQL Server listens on every IP addresses if configured to use TCP/IP.

References:

1.9.4 Dedicated Admin Connection (DAC) Port (Level 1, Scorable)

Description:

Configure Dedicate Admin Connection (DAC) to listen on a fix port.

Rationale:

Using a fix port allows a more reliable connection to the DAC. Also, it allows routing it through firewall if needed.

Remediation:

1. Open Registry Editor.
2. Go to the key `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL10.MSSQLSERVER\MSSQLServer\SuperSocketNetLib\AdminConnection\Tcp` (replace `MSSQLSERVER` with the instance name in case of a named instance).
3. Add the `TcpDynamicPorts` value the desired port number.
4. Restart SQL Server service.
5. Verify if the ERRORLOG contains that DAC is listening on the desired port.

Audit:

Verify if the ERRORLOG contains that DAC is listening on the desired port

Default Value:

By default, DAC listens on to TCP/IP traffic on TCP port 1434 for default instances listening on port 1433 and on random TCP port for in any other cases.

References:

<http://support.microsoft.com/kb/823938/en-us>

1.9.5 Disable SQL Server Browser Service (Level 1, Scorable)

Description:

The SQL Browser service listens for requests on the host level and provides list of available SQL Server instances, information about a particular instance and support for connecting to Dedicated Admin Connection.

This service discloses information about SQL Server installations on a particular host, thus if clients can connect without using it, it should be disabled.

Rationale:

Disabling this service leads to a more secure installation because SQL Server installations cannot be enumerated.

Remediation:

Open a command prompt or powershell window and issue the following command:

```
C:\>sc.exe stop SQLBrowser  
C:\>sc.exe config SQLBrowser start= disabled
```

Audit:

Open a powershell window and issue the following command:

```
PS C:\>Get-WmiObject -Class Win32_Service -Filter "Name='SQLBrowser'"
```

Verify that the following two lines are present in the output:

```
StartMode : Disabled  
State      : Stopped
```

Default Value:

This service is disabled by default if a default instance is installed onto the computer and enabled if a named instance is installed onto the computer.

References:

[http://msdn.microsoft.com/en-us/library/ms181087\(v=SQL.100\).aspx](http://msdn.microsoft.com/en-us/library/ms181087(v=SQL.100).aspx)

1.9.6 Hide SQL Server Instance (Level 2, Scorable)

Description:

SQL Server instance can be configured to not be advertised by the SQL Server Browser service.

Rationale:

In certain cases you may wish to rely on SQL Server Browser service to advertise some of the SQL Server instances installed on the computer while still maintaining a higher level protection for a particular instance.

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 HideInstance box, select Yes, and then click OK to close the dialog box. The change takes effect immediately for new connections.

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 HideInstance 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:

<http://technet.microsoft.com/en-us/library/ms179327.aspx>

1.9.7 Encryption (Level 2, Scorable)

Description:

Microsoft SQL Server can use Secure Sockets Layer (SSL) to encrypt data that is transmitted across a network between an instance of SQL Server and a client application. It is recommended that encryption be forced for all incoming connections. Use the fully-qualified DNS name of the server in the certificate to help prevent masquerading. Do not use a self-signed certificate.

Rationale:

Encrypted connections prevent capturing sensitive data in transit. Using a certificate issued by a CA prevents man-in-the-middle attacks.

Remediation:

Consider implementing SSL connections to SQL Server if you are not already doing so. To set the SQL Server Force Encryption option, perform the following steps:

1. Open SQL Server Configuration Manager.
2. Expand SQL Server Network Configuration and right click Protocols for SQL Server; select Properties.
3. The Force Encryption option may be set on the Flags tab. In order to do so, you must install a suitable certificate first.
4. Restart the SQL Server service.

Audit:

Perform the following steps to determine if SQL Server is using encrypted connections:

1. Open SQL Server Configuration Manager.
2. Expand SQL Server Network Configuration and right click Protocols for SQL Server; select Properties.
3. The Force Encryption option can be reviewed on the Flags tab.

Default Value:

SQL Server connections are not encrypted by default. Note that authentication is encrypted by design and cannot be turned off, therefore login details cannot be captured even if this

recommendation is not followed.

References:

<http://msdn.microsoft.com/en-us/library/ms189067%28v=SQL.100%29.aspx>

1.9.8 Disable Native XML Web Services (Level 1, Scorable)

Description:

Do not configure XML Web Services endpoints where not required. Note that this feature is deprecated and will be removed from a future version of SQL Server.

Rationale:

Native XML Web Services provide database access over HTTP using SOAP messages. Not configuring and activating this feature decreases surface area.

Remediation:

Enumerate SOAP endpoints by selecting from the sys.soap_endpoints catalog view. Use the DROP ENDPOINT command to remove those endpoints.

Audit:

Enumerate SOAP endpoints by selecting from the sys.soap_endpoints catalog view.

Default Value:

This feature is disabled by default in new installations of SQL Server. SQL Server Endpoints are not configured by default.

References:

1.9.9 Database Mirroring Encryption (Level 1, Scorable)

Description:

If Database Mirroring is used, configure AES for transport level encryption.

Rationale:

Encrypting data in transit prevents unauthorized individuals from capturing sensitive data.

Remediation:

Run the following T-SQL query:

```
ALTER ENDPOINT Mirroring
FOR DATABASE_MIRRORING
(
    ENCRYPTION = REQUIRED
    ALGORITHM AES
);
```

Audit:

Run the following T-SQL query:

```
SELECT is_encryption_enabled, encryption_algorithm_desc
FROM sys.database_mirroring_endpoint;
```

Default Value:

By default, encryption is required by database mirroring.

References:

<http://msdn.microsoft.com/en-us/library/ms186360%28v=SQL.100%29.aspx>

1.10 SQL Server Configuration Options

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.

1.10.1 *Ad Hoc Distributed Queries (Level 1, Scorable)*

Description:

By default, SQL Server does not allow ad hoc distributed queries using OPENROWSET and OPENDATASOURCE. When this option is set to 1, SQL Server allows ad hoc access. When this option is not set or is set to 0, SQL Server does not allow ad hoc access.

Rationale:

Enabling the use of ad hoc names means that any authenticated login to SQL Server can access the provider. This can result in allowing users not permitted to access data on the server.

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;
```

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.

Default Value:

By default, this option is disabled.

References:

<http://msdn.microsoft.com/en-us/library/ms187569%28v=SQL.100%29.aspx>

1.10.2 C2 audit mode (Level 2, Scorable)

Description:

C2 audit mode will configure the server to record any attempts to access objects or manipulate permissions. C2 audit mode produces a significant amount of log and if log cannot be written because e.g. the logging partition gets full, SQL Server will shut down. Note that C2 security standard has been superseded by Common Criteria.

Rationale:

C2 audit mode allows detailed tracking of system activity and possible security policy violations. Set this option only when auditing is primary goal.

Remediation:

Run the following T-SQL command:

```
EXECUTE sp_configure 'show advanced options', 1;
RECONFIGURE;
EXECUTE sp_configure 'C2 audit mode', 1;
RECONFIGURE;
GO
EXECUTE sp_configure 'show advanced options', 0;
RECONFIGURE;
```

Restart SQL Server to take the changes into effect.

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 = 'C2 audit mode';
```

Both value columns must show 1.

Default Value:

By default, this option is disabled.

References:

1.10.3 CLR enabled (Level 1, Scorable)

Description:

The CLR enabled option specifies whether user assemblies can be run by SQL Server. If there's no need for it, keep it disabled.

Rationale:

Enabling use of CLR assemblies widens the attack surface of SQL Server and puts it at risk from both inadvertent and malicious assemblies. In case you need CLR, follow the recommendations 1.19.5 and 1.19.6.

Remediation:

Run the following T-SQL command:

```
EXECUTE sp_configure 'clr enabled', 1;  
RECONFIGURE;
```

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.

Default Value:

By default, this option is disabled.

References:

1.10.4 *Common criteria compliance enabled (Level 1, Scorable)*

Description:

This options enables the following elements which are required for the Common Criteria compliance:

- Residual Information Protection
- Ability to view logon statistics
- Column-level GRANT shouldn't override table-level DENY

This option is available in the Enterprise and Developer editions of SQL Server 2008.

Rationale:

Enabling Common Criteria compliance allows a higher level of security for the SQL Server instance.

Remediation:

Run the following T-SQL command:

```
EXECUTE sp_configure 'show advanced options', 1;  
RECONFIGURE;  
EXECUTE sp_configure 'common criteria compliance enabled', 1;  
RECONFIGURE;  
GO  
EXECUTE sp_configure 'show advanced options', 0;  
RECONFIGURE;
```

Restart SQL Server to take the changes into effect.

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 = 'common criteria compliance enabled';
```

Both value columns must show 1.

Default Value:

By default, this option is disabled.

References:

1.10.5 Cross db ownership chaining (Level 1, Scorable)

Description:

This option allows controlling cross-database ownership chaining across all databases. This should be kept disabled.

Rationale:

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.

Remediation:

Run the following T-SQL command:

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

Audit:

Run the following T-SQL command:

```
SELECT name,  
       CAST(value as int) as value_configured,  
       CAST(value_in_use as int) as value_in_use  
FROM sys.configurations  
WHERE name = 'Cross db ownership chaining';
```

Both value columns must show 0.

Default Value:

By default, this option is disabled.

References:

1.10.6 Database Mail XPs (Level 1, Scorable)

Description:

Keep Database Mail XPs disabled, unless you need to send emails from the SQL Server instance.

Rationale:

Disabling Database Mail reduces the SQL Server surface.

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;
```

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.

Default Value:

By default, this option is disabled.

References:

1.10.7 *Default trace enabled (Level 1, Scorable)*

Description:

The default trace provides a troubleshooting and basic security information source to database administrators. This option should be kept enabled all time.

Rationale:

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

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;
```

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.

Default Value:

By default, this option is enabled.

References:**References:**

1.10.8 *Maximum Server Memory (Level 1, Scorable)*

Description:

The min server memory and max server memory configuration options establish lower and upper limits to the amount of memory used by the buffer pool of the Microsoft SQL Server Database Engine. Setting a maximum memory prevents other processes, including the operating system, from being pressed by SQL Server for memory.

Set a maximum memory which leaves enough memory for the operating system and any additional applications running on the server. Reserve 4-8GB memory for the operating system, depending on total physical memory.

Rationale:

Setting a maximum memory prevents other processes, including the operating system, from being pressed by SQL Server for memory.

Remediation:

Run the following T-SQL command:

```
EXECUTE sp_configure 'show advanced options', 1;
RECONFIGURE;
EXECUTE sp_configure 'max server memory', <max_memory>;
RECONFIGURE;
GO
EXECUTE sp_configure 'show advanced options', 0;
RECONFIGURE;
```

Replace <max_memory> with the maximum server memory you want to set, in megabytes.

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 = 'max server memory';
```

Both value columns must show a number less than the total amount of memory in the server, in megabytes.

Default Value:

By default, this option is set to 2,147,483,647.

References:

Comment [BEG12]: Particular suggestion or general?

1.10.9 Ole Automation Procedures (Level 1, Scorable)

Description:

This option defines whether OLE Automation objects can be created via T-SQL batches. Keep this option disabled. Consider offloading such batches from SQL Server.

Rationale:

Enabling this option will increase the attack surface of SQL Server as it will allow instantiation of OLE objects under the security context of SQL Server.

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;
```

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.

Default Value:

By default, this option is disabled.

References:

1.10.10 Remote Access (Level 1, Scorable)

Description:

This option allows EXECUTEution of local stored procedures on remote servers or remote stored procedures on local server. Keep this option disabled unless you need it for replication.

Note that this feature is deprecated.

Rationale:

Leaving this option turned off prevents local stored procedures from being run from a remote server or remote stored procedures from being run on the local server.

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;

```

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.

Default Value:

By default, this option is enabled.

References:

1.10.11 Remote admin connections (Level 1, Scorable)

Description:

This option defines whether the Dedicated Admin Connection (DAC) is listening on localhost only or on the SQL Server IP address. If it's a clustered installation, it 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.

Rationale:

The Dedicated Admin Connection provides a single connection option with reserved resources for a DBA in case the database engine is not responding to normal requests. This should be available only for this purpose.

Remediation:

Run the following T-SQL command on clustered installations:

```

EXECUTE sp_configure 'show advanced options', 1;
RECONFIGURE;
EXECUTE sp_configure 'Remote admin connections', 1;
RECONFIGURE;
GO
EXECUTE sp_configure 'show advanced options', 0;
RECONFIGURE;

```

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 admin connections';
```

Both value columns must show 1 on clustered installations.

Default Value:

By default, this option is disabled.

References:

1.10.12 Scan for startup procs (Level 1, Scorable)

Description:

This option causes SQL Server to scan for and automatically run all stored procedures that are set to execute upon service startup. Ensure that this setting is disabled if not user-defined stored procedures are set to run at startup.

Rationale:

Setting this value to 0 will prevent SQL Server from EXECUTEing startup procedures. This is a defense in depth measure to reduce the threat of an entity leveraging these facilities for malicious purposes. Note: Setting Scan for Startup Procedures to 0 will prevent certain audit traces and other commonly used monitoring SPs from re-starting on start up.

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;
```

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.

Default Value:

By default, this option is disabled.

References:

1.10.13 SQL Mail XPs (Level 1, Scorable)

Description:

Do not enable SQL Mail where not required or where Database Mail could be used instead. Note that SQL Mail is not available on 64-bit installations.

Rationale:

SQL Mail, which is deprecated in favor of Database Mail, relies on a MAPI interface, thus installation and continuous maintenance of a MAPI client, Microsoft Outlook is required for it. This increases the attack surface of the server running the SQL Server instance.

Remediation:

Run the following T-SQL command:

```
EXECUTE sp_configure 'show advanced options', 1;
RECONFIGURE;
EXECUTE sp_configure 'SQL Mail XPs', 0;
RECONFIGURE;
GO
EXECUTE sp_configure 'show advanced options', 0;
RECONFIGURE;
```

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 = 'SQL Mail XPs';
```

Both value columns must show 0.

Default Value:

By default, this option is disabled.

References:

1.10.14 Xp_cmdshell (Level 1, Scorable)

Description:

Disable the xp_cmdshell stored procedure where not required. Consider moving business functions requiring it off from SQL Server or into SQL Agent jobs.

Rationale:

The xp_cmdshell extended stored procedure executes a command string as an operating-system command shell and returns any output as rows of text. If this option is enabled it presents many potential security risks against the SQL Instance and the network it is connected to.

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;
```

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.

Default Value:

By default, this option is disabled.

References:

1.11 General SQL Server Server Level Settings

1.11.1 Database Mail Access (Level 1, Scorable)

Description:

If Database Mail is required and enabled, do not set a public profile and limit access to the function to those logins requiring it.

Rationale:

Access to Database Mail enables a login to send out emails via T-SQL scripts. This can lead to abuse of the Database Mail function, delaying notification emails, applications can send out sensitive data without proper authorization and a malicious user might send false alerts on behalf of the server.

Remediation:

Review if there's any public profile set up on the server. If there's a public profile, replace the permission set to those accounts requiring mail sending capabilities.

1. Add the logins requiring mail sending to the DatabaseMailUserRole role in the msdb database.
2. In SSMS, go to Management / Database Mail. Right click on it and launch Database Mail Configuration Wizard.
3. Select option 2, Manage profile security, click next.
4. On the public profile tab, uncheck the Public option for all profile listed.
5. On the Private Profiles tab, grant the logins access to the appropriate mail profile(s).
6. Click Next and Finish.

Audit:

execute the following T-SQL query:

```
EXECUTE msdb.dbo.sysmail_help_principalprofile_sp;
```

The result set should not contain a record with principal_id = 2.

Default Value:

By default, the Database Mail function is disabled in general.

References:

1.11.2 *Trace Messages (Level 1, Scorable)*

Description:

Do not include EXECUTEution trace messages in the error log output from SQL Server Agent. This setting causes very detailed information to be logged to the SQL Server error logs for each SQL Agent job EXECUTEution.

Rationale:

This is a defense in depth measure to reduce the threat of a disk exhaustion based denial of service.

Remediation:

This setting should only be used in the event that a specific error is being research. Otherwise this setting should be off as it will cause the SQL Server error logs to become very large very quickly.

Audit:

This setting may be viewed on the General Tab of the SQL Server Agent Properties menu. Perform the following steps to view this setting:

1. Open SQL Server Management Studio.
2. Open Object Explorer and connect to the target instance.
3. Expand the instance and right click SQL Server Agent. Select Properties.

The 'Include EXECUTEution trace messages' option may be enabled under the Error Log section on the General tab.

Default Value:

This feature is disabled by default.

References:

<http://msdn.microsoft.com/en-us/library/ms189335.aspx>

<http://msdn.microsoft.com/en-us/library/ms175488.aspx>

1.11.3 *SQL Server Event Forwarding (Level 2, Not Scorable)*

Description:

You can forward to an instance of SQL Server all event messages that meet or exceed a specific error severity level. This is called event forwarding. The forwarding server is a

dedicated server that can also be a master server. You can use event forwarding to centralize alert management for a group of servers, thereby reducing the workload on heavily used servers.

Remediation:

Consider the benefits and drawbacks of event forwarding for your SQL Agent uses. Enabling this feature provides centralized event management, scalability of servers, and efficiency. However, using this feature increases network traffic and introduces a single point of failure for capturing events.

Audit:

To view the status of Event Forwarding in SQL Server, perform the following steps:

1. Open Object Explorer and navigate to SQL Server Agent.
2. Right click SQL Server Agent and select Properties. Navigate to the Advanced tab.
3. The status of event forwarding may be viewed on this tab. Event forwarding may also be configured here.

Default Value:

By default, SQL Agent does not forward events. This feature must be configured.

References:

<http://msdn.microsoft.com/en-us/library/ms189316.aspx>

1.11.4 *Audit Login Attempts (Level 1, Scorable)*

Description:

Through the SQL Server Management Studio, enable auditing login attempts. Consider capturing both failed and successful login attempts.

Rationale:

Maintaining a list of login attempts helps detecting out of policy activities.

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 `Both failed and successful logins` under the "Login Auditing" section and click OK.
4. Restart the SQL Server instance.

Audit:

Perform the following steps to determine the level of auditing currently configured:

1. Open SQL Server Management Studio.
2. Right click the target instance and select Properties and navigate to the Security tab.

3. Verify if the option `Both failed and successful logins` is selected under the “Login Auditing” section.

Default Value:

By default, only failed login attempted are captured.

References:

1.11.5 Do Not Use Remote Servers (Level 1, Scorable)

Description:

Use linked servers rather than remote servers where required.

Rationale:

Remote servers are available for backward compatibility purposes only. Applications that must execute stored procedures against remote instances of SQL Server should use linked servers instead. Note that replication creates remote server registrations; do not modify those if replication is in use otherwise replication may stop working correctly.

Remediation:

Evaluate all remote servers on the instance. Consider removing remote servers and use linked servers when required.

Audit:

Use the following code snippet to find all remote servers on the instance.

```
SELECT *  
FROM sys.servers  
WHERE server_id != 0  
      AND is_linked = 0;
```

Default Value:

Linked servers are not configured by default unless replication is used.

References:

<http://msdn.microsoft.com/en-us/library/ms188279.aspx>

1.11.6 Linked Servers (Level 1, Scorable)

Description:

Configure linked servers to use Windows authentication where required. When linking SQL Server databases, the user’s current identity will be used to authenticate the connection.

Rationale:

The current security context should be used (current user) so that user impersonation is not required at the remote server. This ensures that the user using the linked server needs explicit permissions to perform the intended actions at the target server.

Remediation:

Review the login security context of the linked servers. Ensure that the current security context establishes the connection to the remote server.

Audit:

Use the following code snippet to evaluate the security context of the linked servers on the instance:

```
SELECT * FROM sys.linked_logins l
JOIN sys.server_principals p ON l.local_principal_id = p.principal_id;
```

Default:

No linked servers are defined by default.

References:

<http://msdn.microsoft.com/en-us/library/ms188279%28v=SQL.100%29.aspx>

1.11.7 *sp_processmail (Level 1, Scorable)*

Description:

If you have SQL Mail XPs enabled (see recommendation 1.10.13), avoid using sp_processmail.

Rationale:

The sp_processmail procedure processes information sent to you by mail from unknown sources and can be used to introduce malicious code into your environment. You must take great care to validate the code before EXECUTEing it.

Remediation:

Keep SQL Mail XPs disabled.

Audit:

Verify if SQL Mail XPs are disabled.

Default Value:

By default, this stored procedure is not available because SQL Mail is disabled by default.

References:

<http://msdn.microsoft.com/en-us/library/ms189829%28v=SQL.100%29.aspx>

1.12 SQL Server Database Level Settings

1.12.1 *Encrypt Stored Procedure Definition (Level 2, Not Scorable)*

Description:

If a user-defined stored procedure contains sensitive code or information it should be encrypted.

Rationale:

For stored procedures that contain sensitive information, encrypting them is a way to ensure that others are not able to see the source code. Note that with access to the Dedicated Admin Connection, it is possible in many cases to decrypt the encrypted stored procedure.

Remediation:

Script out the sensitive stored procedures and re-run the scripts as `ALTER PROCEDURE <usp_ProcedureName> WITH ENCRYPTION` option.

Note that if you encrypt a stored procedure, you should keep its clear text definition separately for troubleshooting and development purposes as it cannot be retrieved later.

Audit:

Ensure that all user defined stored procedures containing sensitive data are encrypted. Run the following query to see which procedures are encrypted:

```
SELECT *  
FROM sys.procedures  
WHERE is_encrypted = 1;
```

Default Value:

By default, no stored procedures are encrypted.

References:

<http://msdn.microsoft.com/en-us/library/ms187926%28v=SQL.100%29.aspx>

1.12.2 *DDL Permissions (Level 1, Scorable)*

Description:

DDL statement permissions should only be granted to the database and schema owner, not individual users.

Rationale:

The ability to execute Data Definition Language (DDL) statements to modify database objects should be reserved for database administrators.

Remediation:

Review your DDL permission scheme. Consider revoking DDL permissions from the users. Do not give developers in a production environment these privileges.

Audit:

The following script can be used to determine which users have permissions to alter database objects.

```
SELECT
    ObjectName = o.name,
    ObjectType = o.type_desc,
    PermissionType = dp.state_desc,
    UserType = p.type_desc,
    PermissionName = dp.permission_name,
    UserName = p.name
FROM sys.database_permissions dp
JOIN sys.database_principals p ON dp.grantee_principal_id = p.principal_id
JOIN sys.objects o ON dp.major_id = o.object_id;
```

Default Value:

By default, members of the db_owner database role may modify any object in the database. Schema owners may modify those objects that belong to their schema.

References:

<http://msdn.microsoft.com/en-us/library/ms190387.aspx>

1.12.3 Limit Guest User Permissions (Level 1, Scorable)

Description:

Do not grant the guest user any privileges or permissions unless it is justifiable. Do not add it to predefined database roles.

Rationale:

The guest account allows logins with no explicit access to the database to gain access to it.

Remediation:

Remove the Guest user from any predefined database role. Make sure any permission the user needs are thoroughly tested.

Audit:

Use the following code snippet to determine the permissions explicitly granted to the guest user.

```
SELECT username = p.name, d.*
FROM sys.database_permissions d
JOIN sys.database_principals p ON d.grantee_principal_id = p.principal_id
WHERE p.principal_id = 2;
```

Default Value:

By default, only CONNECT to database permission is granted to the guest user.

References:

1.12.4 Remove Orphan Users (Level 1, Scorable)

Description:

Remove orphan users from databases. Note that if you have users without login intentionally, you should not remove those.

Rationale:

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

Remediation:

Run the following T-SQL query to identify orphan users:

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

Audit:

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

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

Default Value:

N/A

References:

<http://msdn.microsoft.com/en-us/library/ms175475.aspx>

1.12.5 Disable Auto Close Option (Level 1, Scorable)

Description:

Turn off AUTO_CLOSE option for databases.

Rationale:

If a database has AUTO_CLOSE specified, the database files can be accessed on the file system as normal files and can be copied or moved away. This allows a malicious attacker to gain access to data.

Remediation:

If AUTO_CLOSE is allowed, turn it off by running the following T-SQL query:

```
ALTER DATABASE <dbname> SET AUTO_CLOSE = OFF
```

Audit:

Run the following T-SQL query:

```
SELECT name, is_auto_close_on  
FROM sys.databases  
WHERE name != <dbname>;
```

Default Value:

Databases created on SQL Server Enterprise or Standard Editions are not set to auto close. Databases created on SQL Server Express are set to auto close by default.

References:

1.12.6 Use Transparent Data Encryption (TDE) (Level 2, Scorable)

Description:

For databases storing sensitive data, consider implementing Transparent Data Encryption (TDE). Note that TDE is an Enterprise Edition only feature.

Rationale:

TDE encrypts the whole database and prevents retrieving any data from it from outside of the database engine. It prevents abusing backups or shadow copies taken from the database.

Remediation:

Configure TDE for the database, following the article in the References section below.

Audit:

Run the following T-SQL query:

```
SELECT name,  
       is_encrypted  
FROM sys.databases  
WHERE name = <dbname>;
```

Default Value:

By default, no database is encrypted.

References:

<http://msdn.microsoft.com/en-us/library/bb934049%28v=SQL.100%29.aspx>

1.12.7 Turn Off Trustworthy Option (Level 2, Scorable)

Description:

For all databases on the server, turn off the TRUSTWORTHY option, unless required.

Rationale:

The TRUSTWORTHY option allows database objects to access objects in other database under certain circumstances. Limiting this provides protection from malicious CLR assemblies or extended procedures.

Remediation:

Execute the following statement against the database:

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

Audit:

Run the following T-SQL query:

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

Default Value:

By default, no database is encrypted.

References:

<http://msdn.microsoft.com/en-us/library/ms187861%28v=SQL.100%29.aspx>

1.13 Extended stored procedures

Extended stored procedures are to be removed in a future version of SQL Server. They should be avoided at all in new development work or long-term operations. Consider using CLR Integration instead. The following extended stored procedures should not be used by any application or maintenance script. Do not attempt to remove or assign DENY rule to any of these stored procedures. Doing so may result in an unsupported installation of SQL Server 2008.

1.13.1 *User-defined extended stored procedures (Level 1, Scorable)*

Description:

Avoid using user-defined extended stored procedures. If extended functionality is required, use Common Language Runtime (CLR) assemblies instead.

Rationale:

This feature will be removed in a future version of SQL Server.

Remediation:

Convert any user-defined extended stored procedures to CLR stored procedures or consider remove them from the scope of SQL Server.

Audit:

Use the following code snippet to determine if there are user-defined extended procedures on the server:

```
SELECT * FROM master.sys.extended_procedures;
```

Default Value:

By default, there are no user-defined extended procedures on the server.

References:

<http://msdn.microsoft.com/en-us/library/ms187644.aspx>

1.13.2 *xp_availablemedia (Level 1, Scorable)*

Description:

This extended stored procedure is used by SQL Server Management Studio when backing up and restoring databases. This procedure returns a list of available mapped drives to which database backups may be placed.

Rationale:

Ensure this extended stored procedure is not executable to ensure that users are not able to view the local available drives.

Remediation:

Unless required, execute permissions to this extended stored procedure should be revoked from all general users on the SQL Server machine. If enabled, run the following code snippet to revoke execute permissions:

```
REVOKE EXECUTE ON xp_availablemedia TO PUBLIC;
```

Audit:

To view the permissions for this stored procedure, perform the following steps in SQL Server Management Studio:

1. In Object Explorer, navigate to the SQL Server instance and expand the path:
Databases\System Databases\master\Programmability\Extended Stored Procedures
2. Locate `xp_availablemedia`, right click and select Properties.
3. The permissions for this extended stored procedure can be viewed in the Permissions tab.

Default Value:

By default, execute permissions for this extended stored procedure is not given to the public database role.

References:

No references available as this is an undocumented extended stored procedure.

1.13.3 *xp_cmdshell (Level 1, Scorable)*

Description:

This stored procedure spawns a Windows command shell and passes in a string for EXECUTEution.

Rationale:

This stored procedure allows for direct interaction from the SQL Server machine to the operating system. Allowing access to this extended stored procedure potentially allows for malicious operating system and network attacks. This procedure requires CONTROL SERVER permissions.

Comment [BEG13]: FOR CONSENSUS:
Do we want to keep recommendations for undocumented features? That opens a new dimension I think.
Recommendation: drop it.

Remediation:

In most situations, users will not require EXECUTEing this stored procedure. If it is absolutely required, a proxy account can be created to allow for EXECUTEion of this stored procedure by those accounts who are not members of the `sysadmin` sql server group. If this procedure is enabled and not used, ensure that it is disabled.

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;
```

Default Value:

This stored procedure is disabled by default.

References:

<http://msdn.microsoft.com/en-us/library/ms175046.aspx>

1.13.4 *xp_dirtree (Level 1, Scorable)*

Description:

This extended stored procedure returns a result set of the directory tree for a given directory path.

Rationale:

This stored procedure gives insight into the directory structure for the given SQL Server. Ensure that this stored procedure is disabled to ensure that a user is not able discover all folders on the SQL Server.

Remediation:

Unless required, execute permissions to this extended stored procedure should be revoked from all general users on the SQL Server machine. If enabled, run the following code snippet to revoke execute permissions:

```
REVOKE EXECUTE ON xp_fixeddrives TO PUBLIC;
```

Audit:

To view the permissions for this stored procedure, perform the following steps in SQL Server Management Studio:

1. In Object Explorer, navigate to the SQL Server instance and expand the path:
Databases\System Databases\master\Programmability\Extended Stored Procedures
2. Locate `xp_dirtree`, right click and select Properties.
3. The permissions for this extended stored procedure can be viewed in the Permissions tab.

Default Value:

Execute permissions on this stored procedure are denied to the public role by default.

1.13.5 *xp_enumgroups (Level 1, Scorable)*

Description:

This procedure provides a list of local Microsoft Windows groups or a list of global groups that are defined in a specified Windows machine.

Rationale:

To ensure that SQL Server users are not able to determine the groups are present on the SQL Server machine, ensure that execute permissions to this extended stored procedure is denied to the public server group.

Remediation:

To ensure that members of the public server group do not have permissions to execute this stored procedure, execute the following code snippet:

```
REVOKE EXECUTE ON xp_enumgroups to PUBLIC;
```

Audit:

To view the permissions for this stored procedure, perform the following steps in SQL Server Management Studio:

1. In Object Explorer, navigate to the SQL Server instance and expand the path:
Databases\System Databases\master\Programmability\Extended Stored Procedures
2. Locate `xp_enumgroups`, right click and select Properties.
3. The permissions for this extended stored procedure can be viewed in the Permissions tab.

Default Value:

By default, execute permissions for this extended stored procedure is not given to the public database role.

1.13.6 *xp_fixeddrives (Level 1, Scorable)*

Description:

This procedure returns a list of all hard drives on the machine and the space free in megabytes for each drive.

Rationale:

This procedure gives insight into the drives that the SQL Server instance is able to see and the available memory left on these drives. This information could be used for a malicious attack. Ensure that permissions to execute this stored procedure are only given to those users who require it.

Remediation:

Unless required, execute permissions to this extended stored procedure should be revoked from all general users on the SQL Server machine. Run the following code snippet to revoke execute permissions:

```
REVOKE EXECUTE ON xp_fixeddrives TO PUBLIC;
```

Audit:

To view the permissions for this stored procedure, perform the following steps in SQL Server Management Studio:

1. In Object Explorer, navigate to the SQL Server instance and expand the path:
Databases\System Databases\master\Programmability\Extended Stored Procedures
2. Locate `xp_fixeddrives`, right click and select Properties.
3. The permissions for this extended stored procedure can be viewed in the Permissions tab.

Default Value:

By default, the public server group is given execute permissions to this stored procedure.

1.13.7 *xp_getnetname (Level 1, Scorable)*

Description:

This procedure returns the WINS name of the SQL Server machine.

Rationale:

While not extremely useful information, this extended stored procedure should only be executed by system administrators.

Remediation:

Unless required, execute permissions to this extended stored procedure should be revoked from all general users on the SQL Server machine. Run the following code snippet to revoke execute permissions:

```
REVOKE EXECUTE ON xp_getnetname TO PUBLIC;
```

Audit:

To view the permissions for this stored procedure, perform the following steps in SQL Server Management Studio:

1. In Object Explorer, navigate to the SQL Server instance and expand the path:
Databases\System Databases\master\Programmability\Extended Stored Procedures
2. Locate `xp_getnetname`, right click and select Properties.
3. The permissions for this extended stored procedure can be viewed in the Permissions tab.

Default Value:

By default, the public role is given execute permissions to this extended stored procedure.

1.13.8 *xp_logevent (Level 1, Scorable)*

Description:

This procedure logs a user-defined message in the SQL Server log file and in the Windows Event Viewer. `xp_logevent` can be used to send an alert without sending a message to the client.

Rationale:

To prevent error logs from being inundated with erroneous log messages, ensure that the `xp_logevent` extended stored procedure only executable by system administrators.

Remediation:

Unless required, execute permissions to this extended stored procedure should be revoked from all general users on the SQL Server machine. Run the following code snippet to revoke execute permissions:

```
REVOKE EXECUTE ON xp_logevent TO PUBLIC;
```

Audit:

To view the permissions for this stored procedure, perform the following steps in SQL Server Management Studio:

1. In Object Explorer, navigate to the SQL Server instance and expand the path:
Databases\System Databases\master\Programmability\Extended Stored Procedures
2. Locate `xp_logevent`, right click and select Properties.
3. The permissions for this extended stored procedure can be viewed in the Permissions tab.

Default Value:

By default, the public role is not given execute permissions to this extended stored procedure.

1.13.9 *xp_loginconfig (Level 1, Scorable)*

Description:

Reports the login security configuration of an instance of SQL Server when it running on Windows XP or Windows Server 2003/2008.

Rationale:

To prevent general SQL Server users from identifying the SQL Server login security for the given SQL Server instance, ensure that the `xp_loginconfig` extended stored procedure only executable by system administrators.

Remediation:

Unless required, execute permissions to this extended stored procedure should be revoked from all general users on the SQL Server machine. Run the following code snippet to revoke execute permissions:

```
REVOKE EXECUTE ON xp_loginconfig TO PUBLIC;
```

Audit:

To view the permissions for this stored procedure, perform the following steps in SQL Server Management Studio:

1. In Object Explorer, navigate to the SQL Server instance and expand the path:
Databases\System Databases\master\Programmability\Extended Stored Procedures
2. Locate `xp_loginconfig`, right click and select Properties.
3. The permissions for this extended stored procedure can be viewed in the Permissions tab.

Default Value:

By default, the public role is not given execute permissions to this extended stored procedure.

References:

<http://msdn.microsoft.com/en-us/library/ms189514.aspx>

1.13.10 *xp_msver (Level 1, Scorable)*

Description:

Returns version information about Microsoft SQL Server. `xp_msver` also returns information about the actual build number of the server and information about the server environment. The information that `xp_msver` returns can be used within Transact-SQL statements,

batches, stored procedures, and so on, to enhance logic for platform-independent code.

Rationale:

This extended stored procedure returns information regarding system details of the SQL Server instance. This is information you should keep from general SQL Server users. Ensure that the public SQL Server group is not able to execute this stored procedure.

Remediation:

Unless required, execute permissions to this extended stored procedure should be revoked from all general users on the SQL Server machine. Run the following code snippet to revoke execute permissions:

```
REVOKE EXECUTE ON xp_msver TO PUBLIC;
```

Audit:

To view the permissions for this stored procedure, perform the following steps in SQL Server Management Studio:

1. In Object Explorer, navigate to the SQL Server instance and expand the path:
Databases\System Databases\master\Programmability\Extended Stored Procedures
2. Locate `xp_msver`, right click and select Properties.
3. The permissions for this extended stored procedure can be viewed in the Permissions tab.

Default Value:

By default, the public role is given execute permissions to this extended stored procedure.

References:

<http://msdn.microsoft.com/en-us/library/ms187372.aspx>

1.13.11 *xp_readerrorlog (Level 1, Scorable)*

Description:

This extended stored procedure returns a resultset of the values listed in the current error log.

Rationale:

This extended stored procedure returns information regarding events that have been placed in the SQL Server error log. This is critical information that a hacker could potentially use to gauge successful EXECUTEion of scripts. As such, this is information you should keep from general SQL Server users. Ensure that the public SQL Server group is not able to execute this stored procedure.

Remediation:

Unless required, execute permissions to this extended stored procedure should be revoked from all general users on the SQL Server machine. Run the following code snippet to revoke execute permissions:

```
REVOKE EXECUTE ON xp_readerrorlog TO PUBLIC;
```

Audit:

To view the permissions for this stored procedure, perform the following steps in SQL Server Management Studio:

1. In Object Explorer, navigate to the SQL Server instance and expand the path:
Databases\System Databases\master\Programmability\Extended Stored Procedures
2. Locate xp_readerrorlog, right click and select Properties.
3. The permissions for this extended stored procedure can be viewed in the Permissions tab.

Default Value:

By default, the public role is given execute permissions to this extended stored procedure.

References:

No references available as this is an undocumented extended stored procedure.

1.13.12 xp_servicecontrol (Level 1, Scorable)

Description:

This extended stored procedure has the ability to set the runnable status of a service running on the SQL Server machine.

Rationale:

This is a powerful extended stored procedure as it gives the ability to start and stop windows services on the SQL Server machine. Only those users that need to start and stop services should be given this ability, and this ability should be given from outside of the scope of the SQL Server instance. Ensure that this permission is only given to the SQL Server administrator.

Remediation:

Unless required, execute permissions to this extended stored procedure should be revoked from all general users on the SQL Server machine. Run the following code snippet to revoke execute permissions:

```
REVOKE EXECUTE ON xp_servicecontrol TO PUBLIC;
```

Audit:

To view the permissions for this stored procedure, perform the following steps in SQL Server Management Studio:

1. In Object Explorer, navigate to the SQL Server instance and expand the path:
Databases\System Databases\master\Programmability\Extended Stored Procedures
2. Locate `xp_servicecontrol`, right click and select Properties.
3. The permissions for this extended stored procedure can be viewed in the Permissions tab.

Default Value:

By default, the public role is given execute permissions to this extended stored procedure.

References:

No references available as this is an undocumented extended stored procedure.

1.13.13 xp_sprintf (Level 1, Scorable)

Description:

Formats and stores a series of characters and values in the string output parameter. Each format argument is replaced with the corresponding argument.

Rationale:

This is a benign extended stored procedure, but as a matter of consistency should not be executed by general SQL Server users.

Remediation:

Unless required, execute permissions to this extended stored procedure should be revoked from all general users on the SQL Server machine. Run the following code snippet to revoke execute permissions:

```
REVOKE EXECUTE ON xp_sprintf TO PUBLIC;
```

Audit:

To view the permissions for this stored procedure, perform the following steps in SQL Server Management Studio:

1. In Object Explorer, navigate to the SQL Server instance and expand the path:
Databases\System Databases\master\Programmability\Extended Stored Procedures
2. Locate `xp_sprintf`, right click and select Properties.
3. The permissions for this extended stored procedure can be viewed in the Permissions tab.

Default Value:

By default, the public role is given execute permissions to this extended stored procedure.

References:

<http://msdn.microsoft.com/en-us/library/aa260704%28SQL.80%29.aspx>

1.13.14 *xp_sscanf (Level 1, Scorable)*

Description:

Reads data from the string into the argument locations specified by each format argument.

Rationale:

This is a benign extended stored procedure, but as a matter of consistency should not be executed by general SQL Server users.

Remediation:

Unless required, execute permissions to this extended stored procedure should be revoked from all general users on the SQL Server machine. Run the following code snippet to revoke execute permissions:

```
REVOKE EXECUTE ON xp_sscanf TO PUBLIC;
```

Audit:

To view the permissions for this stored procedure, perform the following steps in SQL Server Management Studio:

1. In Object Explorer, navigate to the SQL Server instance and expand the path:
Databases\System Databases\master\Programmability\Extended Stored Procedures
2. Locate `xp_sscanf`, right click and select Properties.
3. The permissions for this extended stored procedure can be viewed in the Permissions tab.

Default Value:

By default, the public role is given execute permissions to this extended stored procedure.

References:

<http://msdn.microsoft.com/en-us/library/ms181431.aspx>

1.13.15 *xp_subdirs (Level 1, Scorable)*

Description:

This extended stored procedure lists all subdirectories listed for a given folder path.

Rationale:

This extended stored procedure gives the user EXECUTEing it insight into all subdirectories on the file system for a given directory path. This is information that a hacker would be able to make use of to determine where key OS files are located. Ensure that only system administrators are able to execute this stored procedure.

Remediation:

Unless required, execute permissions to this extended stored procedure should be revoked from all general users on the SQL Server machine. Run the following code snippet to revoke execute permissions:

```
REVOKE EXECUTE ON xp_subdirs TO PUBLIC;
```

Audit:

To view the permissions for this stored procedure, perform the following steps in SQL Server Management Studio:

1. In Object Explorer, navigate to the SQL Server instance and expand the path:
Databases\System Databases\master\Programmability\Extended Stored Procedures
2. Locate `xp_subdirs`, right click and select Properties.
3. The permissions for this extended stored procedure can be viewed in the Permissions tab.

Default Value:

By default, the public role is not given execute permissions to this extended stored procedure.

References:

No references available as this is an undocumented extended stored procedure.

1.13.16 *xp_regaddmultistring (Level 1, Scorable)*

Description:

This extended stored procedure is used to add multiple strings to the server's registry. Limit EXECUTE permission to the sysadmin role only.

Rationale:

SQL Server logins not performing administrative tasks have no need to modify the registry via SQL Server.

Remediation:

To ensure that general SQL Server users are not able to execute this stored procedure, execute the following code snippet:

```
REVOKE EXECUTE ON xp_regaddmultistring TO PUBLIC;
```

Audit:

To view the permissions for this stored procedure, perform the following steps in SQL Server Management Studio:

1. In Object Explorer, navigate to the SQL Server instance and expand the path:
Databases\System Databases\master\Programmability\Extended Stored Procedures
2. Locate `xp_regaddmultistring`, right click and select Properties.

3. The permissions for this extended stored procedure can be viewed in the Permissions tab.

Default Value:

execute permissions are not revoked on this system stored procedure by default.

References:

This stored procedure is undocumented and thus no official reference exists.

1.13.17 xp_regdeletekey (Level 1, Scorable)

Description:

This extended stored procedure is used to delete registry keys from the server's registry. Limit EXECUTE permission to the sysadmin role only.

Rationale:

SQL Server logins not performing administrative tasks have no need to modify the registry via SQL Server.

Remediation:

To ensure that general SQL Server users are not able to execute this stored procedure, execute the following code snippet:

```
REVOKE EXECUTE ON xp_regdeletekey TO PUBLIC;
```

Audit:

To view the permissions for this stored procedure, perform the following steps in SQL Server Management Studio:

1. In Object Explorer, navigate to the SQL Server instance and expand the path:
Databases\System Databases\master\Programmability\Extended Stored Procedures
2. Locate `xp_regdeletekey`, right click and select Properties.
3. The permissions for this extended stored procedure can be viewed in the Permissions tab.

Default Value:

execute permissions are not revoked on this system stored procedure by default.

References:

This stored procedure is undocumented and thus no official reference exists.

1.13.18 xp_regdeletevalue (Level 1, Scorable)

Description:

This extended stored procedure is used to delete values from the server's registry. Limit EXECUTE permission to the sysadmin role only.

Rationale:

SQL Server logins not performing administrative tasks have no need to modify the registry via SQL Server.

Remediation:

To ensure that general SQL Server users are not able to execute this stored procedure, execute the following code snippet:

```
REVOKE EXECUTE ON xp_regdeletevalue TO PUBLIC;
```

Audit:

To view the permissions for this stored procedure, perform the following steps in SQL Server Management Studio:

1. In Object Explorer, navigate to the SQL Server instance and expand the path:
Databases\System Databases\master\Programmability\Extended Stored Procedures
2. Locate `xp_regdeletevalue`, right click and select Properties.
3. The permissions for this extended stored procedure can be viewed in the Permissions tab.

Default Value:

execute permissions are not revoked on this system stored procedure by default.

References:

This stored procedure is undocumented and thus no official reference exists.

1.13.19 *xp_regenumvalues (Level 1, Scorable)*

Description:

This extended stored procedure is used to enumerate a set of values in a registry path. Limit EXECUTE permission to the sysadmin role only.

Rationale:

SQL Server logins not performing administrative tasks have no need to read the registry via SQL Server.

Remediation:

To ensure that general SQL Server users are not able to execute this stored procedure, execute the following code snippet:

```
REVOKE EXECUTE ON xp_regenumvalues TO PUBLIC;
```

Audit:

To view the permissions for this stored procedure, perform the following steps in SQL Server Management Studio:

1. In Object Explorer, navigate to the SQL Server instance and expand the path:
Databases\System Databases\master\Programmability\Extended Stored Procedures
2. Locate `xp_regenumvalues`, right click and select Properties.

3. The permissions for this extended stored procedure can be viewed in the Permissions tab.

Default Value:

execute permissions are not revoked on this system stored procedure by default.

References:

This stored procedure is undocumented and thus no official reference exists.

1.13.20 *xp_regremovemultistring (Level 1, Scorable)*

Description:

This extended stored procedure is used to remove multiple strings from the server's registry. Limit EXECUTE permission to the sysadmin role only.

Rationale:

SQL Server logins not performing administrative tasks have no need to modify the registry via SQL Server.

Remediation:

To ensure that general SQL Server users are not able to execute this stored procedure, execute the following code snippet:

```
REVOKE EXECUTE ON xp_regremovemultistring TO PUBLIC;
```

Audit:

To view the permissions for this stored procedure, perform the following steps in SQL Server Management Studio:

1. In Object Explorer, navigate to the SQL Server instance and expand the path:
Databases\System Databases\master\Programmability\Extended Stored Procedures
2. Locate `xp_regremovemultistring`, right click and select Properties.
3. The permissions for this extended stored procedure can be viewed in the Permissions tab.

Default Value:

execute permissions are not revoked on this system stored procedure by default.

References:

This stored procedure is undocumented and thus no official reference exists.

1.13.21 *xp_regwrite (Level 1, Scorable)*

Description:

This extended stored procedure is used to write key values to the server's registry. Limit EXECUTE permission to the sysadmin role only.

Rationale:

SQL Server logins not performing administrative tasks have no need to write the registry via SQL Server.

Remediation:

To ensure that general SQL Server users are not able to execute this stored procedure, execute the following code snippet:

```
REVOKE EXECUTE ON xp_regwrite TO PUBLIC;
```

Audit:

To view the permissions for this stored procedure, perform the following steps in SQL Server Management Studio:

1. In Object Explorer, navigate to the SQL Server instance and expand the path:
Databases\System Databases\master\Programmability\Extended Stored Procedures
2. Locate `xp_regwrite`, right click and select Properties.
3. The permissions for this extended stored procedure can be viewed in the Permissions tab.

Default Value:

execute permission is not granted to the public role by default.

References:

This stored procedure is undocumented and thus no official reference exists.

1.13.22 *xp_regread (Level 1, Scorable)*

Description:

This extended stored procedure is used to read key values from the server's registry. Limit EXECUTE permission to the sysadmin role only.

Rationale:

SQL Server logins not performing administrative tasks have no need to read the registry via SQL Server.

Remediation:

To ensure that general SQL Server users are not able to execute this stored procedure, execute the following code snippet:

```
REVOKE EXECUTE ON xp_regread TO PUBLIC;
```

Audit:

To view the permissions for this stored procedure, perform the following steps in SQL Server Management Studio:

4. In Object Explorer, navigate to the SQL Server instance and expand the path:
Databases\System Databases\master\Programmability\Extended Stored Procedures
5. Locate `xp_regwrite`, right click and select Properties.

6. The permissions for this extended stored procedure can be viewed in the Permissions tab.

Default Value:

execute permission is granted to the public role by default.

References:

This stored procedure is undocumented and thus no official reference exists.

1.14 Authentication and Authorization

1.14.1 *Authentication mode (Level 1, Scorable)*

Description:

When possible, the SQL instance should be in Windows-only authentication mode. Mixed-mode authentication should only be used in those situations where it is absolutely necessary.

Rationale:

Windows provides a more robust authentication mechanism than SQL Server authentication. If SQL Server authentication is required, configure SQL Server account password and lockout properties with local or domain-based group policies.

Remediation:

If SQL Server authentication is enabled but not required, switch to Windows-only authentications.

Audit:

Perform the following steps to determine the current SQL Server Authentication mode:

1. Open SQL Server Management Studio.
2. Open the Object Explorer tab and connect to the target database instance. If you use a SQL Server username and password to connect, then you may omit the following steps. Your authentication mode is Mixed Mode. If you connect using Windows authentication, please continue.
3. Right click the instance name and select Properties.
4. Select the Security page from the left menu.
5. The authentication mode is listed under Server Authentication.

Default Value:

By default, SQL Server is installed with Windows-only authentication.

References:

1.14.2 *Shared Accounts (Level 1, Scorable)*

Description:

Avoid using shared accounts on SQL Server. Create dedicated accounts to every person requiring it. Discourage sharing personalized accounts with other persons.

Rationale:

Having personalized accounts allows more adequate permission granting and supports auditing.

Remediation:

Remove any account used by multiple persons from the system and replace them with personalized accounts.

Audit:

List the accounts on the SQL Server. Ensure that all accounts are personalized.

Default Value:

Not Applicable

References:

1.14.3 *Windows Logins (Level 1, Scorable)*

Description:

Grant persons permission to the SQL Server via their Windows account if possible. Avoid creating SQL login for those persons.

Rationale:

Using Windows accounts provides more security.

Remediation:

Review SQL logins and switch them to Windows logins if possible.

Audit:

Review SQL logins and verify if they cannot be switched to Windows logins.

Default Value:

By default, SQL Server installs with Windows only authentication.

References:

1.14.4 *SQL Logins (Level 1, Not Scorable)*

Description:

If personalized SQL logins are provided to users, enforce those accounts to change password on next logon after the login details are handed over to the account owner.

Rationale:

Enforcing password change will prevent the account administrators or anyone accessing the initial password to misuse the SQL login created without being noticed.

Remediation:

When a new login is created, use the MUST_CHANGE option in the CREATE USER statement.

Audit:

Review the account creation procedures.

Default Value:

By default, this option is not set.

References:

1.14.5 Assigning System Administrators role (Level 1, Not Scorable)

Description:

When assigning database administrators to the System Administrators role, grant them the privilege via their Windows accounts. Assign only authorized DBAs to the SQL Server System Administrators role.

Rationale:

Only those individuals who should perform administrative tasks should have assigned to the `sysadmin` (sa) system role. In SQL 2008, there are many additional roles and privileges, aimed to minimize the number of logins in the `sysadmin` role.

Remediation:

Revoke `sysadmin` privileges to any account that does not perform administrative duties that require it. Assign them specific privileges instead.

Audit:

Use the following code snippet to determine those logins that belong to the `sysadmin` server role:

```
SELECT loginname = p.name, logintype = p.type_desc
FROM sys.server_role_members m
JOIN sys.server_principals p ON m.member_principal_id = p.principal_id
JOIN sys.server_principals r ON m.role_principal_id = r.principal_id
WHERE r.name = 'sysadmin';
```

Default Value:

By default, there are no members in the `sysadmin` fixed server role. At installation time, the person performing the installation should explicitly specify the role members.

References:

<http://msdn.microsoft.com/en-us/library/ms188659%28v=SQL.100%29.aspx>

1.14.6 *Ensure SQL Logins have a Strong Password (Level 1, Not Scorable)*

Description:

Ensure that all SQL Logins have strong passwords.

Rationale:

Verify that the passwords are not blank and cannot be easily compromised. Ensure that the `CHECK_POLICY` option is enabled for the SQL Login. This option ensures that the SQL login follows the same password complexity policies as the Operating System.

Remediation:

Ensure that the `CHECK_POLICY` option is enabled for all SQL logins.

Audit:

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

```
SELECT SQLLoginName = sp.name,  
       PasswordPolicyEnforced = CAST(sl.is_policy_checked AS BIT)  
FROM sys.server_principals sp  
JOIN sys.sql_logins AS sl ON sl.principal_id = sp.principal_id  
WHERE sp.type_desc = 'SQL_LOGIN';
```

Default Value:

`Check_Policy` is enforced by default for SQL Server logins.

References:

<http://msdn.microsoft.com/en-us/library/ms161959.aspx>

1.14.7 *Fixed Server Roles (Level 1, Not Scorable)*

Description:

Only use the fixed server roles `sysadmin`, `securityadmin`, `serveradmin`, `setupadmin` etc, to support DBA activity.

Description:

Avoid assigning these roles to application database user accounts, application administrator accounts, application developer accounts or application roles.

Rationale:

Only database administrators should have elevated permissions. Application accounts, developers, and other non-admin roles should only be assigned the permissions necessary to interact with the database to perform their duties.

Remediation:

Remove role membership for those accounts that do not complete regular administrative activity.

Audit:

Run the following code snippet to determine server role members:

```
SELECT loginname = p.name, serverrolename = r.name, logintype = p.type_desc
FROM sys.server_role_members m
JOIN sys.server_principals p ON m.member_principal_id = p.principal_id
JOIN sys.server_principals r ON m.role_principal_id = r.principal_id;
```

Default Value:

New logins are not members of the server roles by default.

References:

<http://msdn.microsoft.com/en-us/library/ms188659.aspx>

1.14.8 Database Roles (Level 1, Scorable)

Description:

Only use the fixed database role db_owner to support DBA activity.

Description:

Avoid assigning this role to application database user accounts, application administrator accounts, application developer accounts or application roles.

Rationale:

Only database administrators should have elevated permissions. Application accounts, developers, and other non-admin roles should only be assigned the permissions necessary to interact with the database to perform their duties.

Remediation:

Remove role membership for those accounts that do not complete regular administrative activity.

Audit:

Run the following code snippet to determine database role members:

```
SELECT loginname = s.name, username = p.name, dbrolename = r.name, logintype
= p.type_desc
FROM sys.database_role_members m
JOIN sys.database_principals p ON m.member_principal_id = p.principal_id
JOIN sys.database_principals r ON m.role_principal_id = r.principal_id
JOIN sys.server_principals s ON s.sid = p.sid;
```

Default Value:

New users are not members of database roles by default.

References:

1.14.9 Guest Database User (Level 1, Scorable)

Description:

The guest account may allow a login without a user account to access a database. 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. The guest user account should be denied `CONNECT` permissions in all databases except master and tempdb.

Rationale:

The guest account cannot be revoked `CONNECT` permissions in master and tempdb, but should be revoked in all other databases on the SQL Server instance. This ensures that a login is not able to access database information without explicit access to do so.

Remediation:

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

```
REVOKE CONNECT FROM guest;
```

Audit:

Run the following code snippet in each database in the instance to determine if the guest user exists.

```
SELECT DB_NAME(), name
FROM sys.database_principals
WHERE name = 'guest'
```

Default Value:

The guest user account is added to each new database by default.

References:

<http://msdn.microsoft.com/en-us/library/bb402861%28v=sql.100%29.aspx>

1.14.10 Using the GRANT option (Level 1, Scorable)

Description:

The grant option allows the grantee the ability to grant specific permissions to other principals.

Rationale:

Do not assign the `GRANT` option of object permission to a user or role. Assigning this permission gives away an element of security to the user to which the permission is assigned.

Remediation:

If you uncover a situation where the `GRANT` option has been given to a user in a database, determine if this user requires this permission. If they do not, revoke this permission. If they do need it, consider creating a separate role and grant the role the `GRANT` permission.

Audit:

Use the following code snippet to determine the users who have been explicitly given the `GRANT` option:

```
SELECT ObjectName = o.name,  
       ObjectType = o.type_desc,  
       PermissionType = dp.state_desc,  
       UserType = p.type_desc,  
       PermissionName = dp.permission_name,  
       UserName = p.name, *  
FROM sys.database_permissions dp  
JOIN sys.database_principals p ON dp.grantee_principal_id = p.principal_id  
JOIN sys.objects o ON dp.major_id = o.object_id  
WHERE dp.state = 'W';
```

Default Value:

No user account is explicitly given the `GRANT` option by default.

References:

<http://msdn.microsoft.com/en-us/library/ms187965%28v=SQL.100%29.aspx>

1.14.11 User-defined Database Roles (Level 2, Not Scorable)

Description:

Create user-defined database roles to assign permissions to objects in the database when a pre-defined database role does not supply the appropriate permissions to a group of users.

Rationale:

Not all organizations have a need for user-defined database roles. This may not apply to all organizations.

Remediation:

Consider using user-defined database roles in your SQL Server security scheme if you do not already do so.

Audit:

Use the following code snippet to identify all user-defined database roles in the current database.

```
SELECT name  
FROM sys.database_principals  
WHERE type = 'R';
```

Default Value:

Besides public, no user-defined database roles exist by default.

References:

1.14.12 Role Assignments (Level 1, Not Scorable)

Description:

Assign permissions to roles rather than users. The principle of “Least Privilege” applies, thus users should not be given access to roles they do not need for their job function.

Rationale:

Ensure that roles, rather than users own objects to avoid application changes when a user is dropped. Ensure that roles have `EXECUTE` permissions to stored procedures.

Remediation:

Consider using a database role security scheme if you are not currently doing so.

Audit:

Run the following snippet to determine database permissions not assigned specifically to a database role.

```
SELECT dp.permission_name,  
       dp.state_desc,  
       p.name,  
       p.type_desc  
FROM sys.database_permissions dp  
JOIN sys.database_principals p ON dp.grantee_principal_id = p.principal_id  
JOIN sys.objects o ON dp.major_id = o.object_id  
WHERE p.type_desc <> 'DATABASE_ROLE';
```

Default Value:

By default, specific permissions are not assigned to database users and no custom roles are created.

References:

1.15 Auditing and Logging

TODO: Pair SQL Trace and SQL Audit events; break them into two parts, level 1 and 2.

1.15.1 Review Collected Audit Log (Level 1, Not Scorable)

Description:

Prepare a schedule for reviewing audit log collected based on section 1.16 regularly.

Rationale:

Regular audit log reviews should be conducted to identify any attempted security breach.

Remediation:

Begin performing scheduled reviews of SQL Server auditing data throughout the year. This information should identify any glaring attempts by an outsider to breach the SQL Server.

Audit:

Begin the audit by reviewing the SQL Server error logs and any predefined server side traces that run in the background. This should include the default SQL Server trace.

Default Value:

Not Applicable

References:

1.15.2 Secure Audit Files (Level 1, Not Scorable)

Description:

Secure the audit files generated based on section 1.16 so that only authorized accounts can access it.

Rationale:

The audit log may contain sensitive data; limiting access to it to authorized accounts prevents information disclosure.

Remediation:

Review the NTFS permissions on the directory storing the audit files; remove users and groups not requiring access to it.

Audit:

Review the NTFS permissions on the directory storing the audit files.

Default Value:

Not Applicable

References:

1.15.3 SQL Server Logs (Level 1, Scorable)

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.

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. The number of SQL Error logs may be configured from this menu.

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. The number of SQL Error logs can be seen.

Default Value:

The default number of SQL Server logs kept is 6. After 6 files have been reached, recycling occurs.

References:

1.15.4 Review SQL Permissions Granted (Level 1, Scorable)

Description:

Regularly review permissions granted in SQL Server – see sys.database_permissions and sys.server_permissions.

Rationale:

Reviewing permissions regularly ensures that permissions not needed anymore are revoked and allows detection of out-of-place permissions.

Remediation:

Schedule regular permission reviews.

Audit:

Review permissions granted by listing the content of the sys.database_permissions and sys.server_permissions catalog views.

Default Value:

Not Applicable.

References:

1.15.5 Periodic scan of stored procedures (Level 1, Not Scorable)

Description:

Verify stored procedures that have been set to AutoStart are secure.

Rationale:

Stored procedures that are set to run when SQL Server starts should be periodically evaluated to ensure that malicious code has not been injected into the code.

Remediation:

Verify the coding of any procedure that are set to AutoStart. Verify if they are actually necessary to set to autostart. Consider setting up SQL Server Agent jobs running at agent startup instead. These procedures are located in the master db and are executed when SQL Server starts.

Audit:

The following code snippet can be ran to find procedures that are set to AutoStart.

```
SELECT name
FROM master.sys.procedures
WHERE is_auto_executed = 1
```

Default:

User-defined stored procedures are not set to AutoStart by default.

References:

<http://msdn.microsoft.com/en-us/library/ms188737%28v=SQL.100%29.aspx>

1.16 Audit SQL Trace Configuration

1.16.1 SQL Trace Setup (Level 1, Scorable)

Description:

In order to maintain a detailed audit trace of ongoing activities in SQL Server, create a trace and add the events and data columns described in this section later to it. Consider creating a different trace than the default trace to ensure that in case an excessive amount of data is logged and the audit trace jeopardizes the stability of the system, a basic set of data can still be collected about most critical activities.

Rationale:

Collecting detailed audit information allows adequate investigation of any potential or actual security issues.

Remediation:

Set up a SQL trace by using the sp_trace_create stored procedure.

Audit:

Ensure that the trace is running by querying the running traces with the following T-SQL query:

```
SELECT *
FROM sys.fn_trace_getinfo(0)
WHERE property in (2,5);
```

This will return two rows per trace: property = 2 tells the file name and property = 5 tells the state: if it's running, value = 1.

Default Value:

By default, there is only a default trace running on SQL Server.

References:

1.16.2 *Audit Add Login to Server Role Event Class (Level 1, Scorable)*

Description:

Add the Audit Add Login to Server Role Event Class to the audit trace.

Rationale:

This event class indicates that a login was added or removed from a fixed server role.

Remediation:

Stop your trace if running and add the Audit Add Login to Server Role Event Class to it:

```
-- Identify your trace with the following query:
SELECT * FROM sys.fn_trace_getinfo(0) WHERE property = 2;

DECLARE @traceid int = <set your traceid here>;
--stop the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 0;
--add the new event class
EXECUTE sp_trace_setevent @TraceID, 108, 1, 1; -- TextData
EXECUTE sp_trace_setevent @TraceID, 108, 11, 1; -- LoginName
EXECUTE sp_trace_setevent @TraceID, 108, 14, 1; -- StartTime
EXECUTE sp_trace_setevent @TraceID, 108, 21, 1; -- EventSubClass
EXECUTE sp_trace_setevent @TraceID, 108, 23, 1; -- Success
EXECUTE sp_trace_setevent @TraceID, 108, 35, 1; -- DatabaseName
EXECUTE sp_trace_setevent @TraceID, 108, 38, 1; -- RoleName
EXECUTE sp_trace_setevent @TraceID, 108, 40, 1; -- DBUserName
EXECUTE sp_trace_setevent @TraceID, 108, 64, 1; -- SessionLoginName
--start the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 1;
```

Audit:

Verify the events captured by the trace:

Run the T-SQL query below:

```
SELECT
    CAST(e.name as varchar(70)) AS Event_Descr,
    CAST(c.name as varchar(70)) AS Column_Descr
FROM sys.fn_trace_geteventinfo(1) t
    INNER JOIN sys.trace_events e
        ON t.eventID = e.trace_event_id
    INNER JOIN sys.trace_columns c
        ON t.columnid = c.trace_column_id
WHERE e.name = 'Audit Add Login to Server Role Event'
```



```
ORDER BY e.name;
```

This query should return the columns listed in the Remediation section.

Default Value:

By default, this information is captured in the default trace.

References:

<http://www.microsoft.com/sqlserver/en/us/common-criteria.aspx>

1.16.3 Audit Add Member to DB Role Event Class (Level 1, Scorable)

Description:

Add the Audit Add Member to DB Role Event Class to the audit trace.

Rationale:

This event class indicates that a login has been added to or removed from a role.

Remediation:

Stop your trace if running and add the Audit Add Member to DB Role Event Class to it:

```
-- Identify your trace by file name with the following query:
SELECT * FROM sys.fn_trace_getinfo(0) WHERE property = 2;

DECLARE @traceid int = <set your traceid here>;
--stop the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 0;
--add the new event class
EXECUTE sp_trace_setevent @TraceID, 110, 1, 1; -- TextData
EXECUTE sp_trace_setevent @TraceID, 110, 11, 1; -- LoginName
EXECUTE sp_trace_setevent @TraceID, 110, 14, 1; -- StartTime
EXECUTE sp_trace_setevent @TraceID, 110, 21, 1; -- EventSubClass
EXECUTE sp_trace_setevent @TraceID, 110, 23, 1; -- Success
EXECUTE sp_trace_setevent @TraceID, 110, 35, 1; -- DatabaseName
EXECUTE sp_trace_setevent @TraceID, 110, 38, 1; -- RoleName
EXECUTE sp_trace_setevent @TraceID, 110, 40, 1; -- DBUserName
EXECUTE sp_trace_setevent @TraceID, 110, 64, 1; -- SessionLoginName
--start the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 1;
```

Audit:

Verify the events captured by the trace:

Run the T-SQL query below:

```
SELECT
    CAST(e.name as varchar(70)) AS Event_Descr,
    CAST(c.name as varchar(70)) AS Column_Descr
FROM sys.fn_trace_geteventinfo(1) t
    INNER JOIN sys.trace_events e
        ON t.eventID = e.trace_event_id
    INNER JOIN sys.trace_columns c
        ON t.columnid = c.trace_column_id
WHERE e.name = 'Audit Add Member to DB Role Event'
```

```
ORDER BY e.name;
```

This query should return the columns listed in the Remediation section.

Default Value:

By default, this information is captured in the default trace.

References:

<http://www.microsoft.com/sqlserver/en/us/common-criteria.aspx>

1.16.4 Audit App Role Change Password Event Class (Level 1, Scorable)

Description:

Add the Audit App Role Change Password Event Class to the audit trace.

Rationale:

This event class indicates that a password has been changed for an application role.

Remediation:

Stop your trace if running and add the Audit App Role Change Password Event Class to it:

```
-- Identify your trace by file name with the following query:
SELECT * FROM sys.fn_trace_getinfo(0) WHERE property = 2;

DECLARE @traceid int = <set your traceid here>;
--stop the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 0;
--add the new event class
EXECUTE sp_trace_setevent @TraceID, 112, 1, 1; -- TextData
EXECUTE sp_trace_setevent @TraceID, 112, 11, 1; -- LoginName
EXECUTE sp_trace_setevent @TraceID, 112, 14, 1; -- StartTime
EXECUTE sp_trace_setevent @TraceID, 112, 23, 1; -- Success
EXECUTE sp_trace_setevent @TraceID, 112, 34, 1; -- ObjectName
EXECUTE sp_trace_setevent @TraceID, 112, 35, 1; -- DatabaseName
EXECUTE sp_trace_setevent @TraceID, 112, 40, 1; -- DBUserName
EXECUTE sp_trace_setevent @TraceID, 112, 64, 1; -- SessionLoginName
--start the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 1;
```

Audit:

Verify the events captured by the trace:

Run the T-SQL query below:

```
SELECT
    CAST(e.name as varchar(70)) AS Event_Descr,
    CAST(c.name as varchar(70)) AS Column_Descr
FROM sys.fn_trace_geteventinfo(1) t
    INNER JOIN sys.trace_events e
        ON t.eventID = e.trace_event_id
    INNER JOIN sys.trace_columns c
        ON t.columnid = c.trace_column_id
```

```
WHERE e.name = 'Audit App Role Change Password Event'
ORDER BY e.name;
```

This query should return the columns listed in the Remediation section.

Default Value:

By default, this information is not captured in the default trace.

References:

<http://www.microsoft.com/sqlserver/en/us/common-criteria.aspx>

1.16.5 Audit Backup/Restore Event Class (Level 1, Scorable)

Description:

Add the Audit Backup/Restore Event Class to the audit trace.

Rationale:

This event class indicates that a backup or restore statement has been issued.

Remediation:

Stop your trace if running and add the Audit Backup/Restore Event Class to it:

```
-- Identify your trace by file name with the following query:
SELECT * FROM sys.fn_trace_getinfo(0) WHERE property = 2;

DECLARE @traceid int = <set your traceid here>;
--stop the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 0;
--add the new event class
EXECUTE sp_trace_setevent @TraceID, 115, 1, 1; -- TextData
EXECUTE sp_trace_setevent @TraceID, 115, 11, 1; -- LoginName
EXECUTE sp_trace_setevent @TraceID, 115, 14, 1; -- StartTime
EXECUTE sp_trace_setevent @TraceID, 115, 21, 1; -- EventSubClass
EXECUTE sp_trace_setevent @TraceID, 115, 23, 1; -- Success
EXECUTE sp_trace_setevent @TraceID, 115, 34, 1; -- ObjectName
EXECUTE sp_trace_setevent @TraceID, 115, 35, 1; -- DatabaseName
EXECUTE sp_trace_setevent @TraceID, 115, 40, 1; -- DBUserName
EXECUTE sp_trace_setevent @TraceID, 115, 64, 1; -- SessionLoginName
--start the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 1;
```

Audit:

Verify the events captured by the trace:

Run the T-SQL query below:

```
SELECT
    CAST(e.name as varchar(70)) AS Event_Descr,
    CAST(c.name as varchar(70)) AS Column_Descr
FROM sys.fn_trace_geteventinfo(1) t
    INNER JOIN sys.trace_events e
        ON t.eventID = e.trace_event_id
    INNER JOIN sys.trace_columns c
        ON t.columnid = c.trace_column_id
```

```
WHERE e.name = 'Audit Backup/Restore Event'
ORDER BY e.name;
```

This query should return the columns listed in the Remediation section.

Default Value:

By default, this information is captured in the default trace.

References:

<http://www.microsoft.com/sqlserver/en/us/common-criteria.aspx>

1.16.6 Audit Change Audit Event Class (Level 1, Scorable)

Description:

Add the Audit Change Audit Event Class to the audit trace.

Rationale:

This event class indicates that an audit trace modification has been made.

Remediation:

Stop your trace if running and add the Audit Change Audit Event Class to it:

```
-- Identify your trace by file name with the following query:
SELECT * FROM sys.fn_trace_getinfo(0) WHERE property = 2;

DECLARE @traceid int = <set your traceid here>;
--stop the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 0;
--add the new event class
EXECUTE sp_trace_setevent @TraceID, 117, 1, 1; -- TextData
EXECUTE sp_trace_setevent @TraceID, 117, 11, 1; -- LoginName
EXECUTE sp_trace_setevent @TraceID, 117, 14, 1; -- StartTime
EXECUTE sp_trace_setevent @TraceID, 117, 21, 1; -- EventSubClass
EXECUTE sp_trace_setevent @TraceID, 117, 23, 1; -- Success
EXECUTE sp_trace_setevent @TraceID, 117, 35, 1; -- DatabaseName
EXECUTE sp_trace_setevent @TraceID, 117, 40, 1; -- DBUserName
EXECUTE sp_trace_setevent @TraceID, 117, 64, 1; -- SessionLoginName
--start the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 1;
```

Audit:

Verify the events captured by the trace:

Run the T-SQL query below:

```
SELECT
    CAST(e.name as varchar(70)) AS Event_Descr,
    CAST(c.name as varchar(70)) AS Column_Descr
FROM sys.fn_trace_geteventinfo(1) t
    INNER JOIN sys.trace_events e
        ON t.eventID = e.trace_event_id
    INNER JOIN sys.trace_columns c
        ON t.columnid = c.trace_column_id
WHERE e.name = 'Audit Change Audit Event'
```

```
ORDER BY e.name;
```

This query should return the columns listed in the Remediation section.

Default Value:

By default, this information is captured in the default trace.

References:

<http://www.microsoft.com/sqlserver/en/us/common-criteria.aspx>

1.16.7 Audit Change Database Owner Event Class (Level 1, Scorable)

Description:

Add the Audit Change Database Owner Event Class to the audit trace.

Rationale:

This event class indicates that the permissions to change the owner of a database have been checked.

Remediation:

Stop your trace if running and add the Audit Change Database Owner Event Class to it:

```
-- Identify your trace by file name with the following query:
SELECT * FROM sys.fn_trace_getinfo(0) WHERE property = 2;

DECLARE @traceid int = <set your traceid here>;
--stop the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 0;
--add the new event class
EXECUTE sp_trace_setevent @TraceID, 152, 1, 1; -- TextData
EXECUTE sp_trace_setevent @TraceID, 152, 11, 1; -- LoginName
EXECUTE sp_trace_setevent @TraceID, 152, 14, 1; -- StartTime
EXECUTE sp_trace_setevent @TraceID, 152, 23, 1; -- Success
EXECUTE sp_trace_setevent @TraceID, 152, 34, 1; -- ObjectName
EXECUTE sp_trace_setevent @TraceID, 152, 35, 1; -- DatabaseName
EXECUTE sp_trace_setevent @TraceID, 152, 40, 1; -- DBUserName
EXECUTE sp_trace_setevent @TraceID, 152, 64, 1; -- SessionLoginName
--start the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 1;
```

Audit:

Verify the events captured by the trace:

Run the T-SQL query below:

```
SELECT
    CAST(e.name as varchar(70)) AS Event_Descr,
    CAST(c.name as varchar(70)) AS Column_Descr
FROM sys.fn_trace_geteventinfo(1) t
    INNER JOIN sys.trace_events e
        ON t.eventID = e.trace_event_id
    INNER JOIN sys.trace_columns c
        ON t.columnid = c.trace_column_id
WHERE e.name = 'Audit Change Database Owner Event'
```

```
ORDER BY e.name;
```

This query should return the columns listed in the Remediation section.

Default Value:

By default, this information is captured in the default trace.

References:

<http://www.microsoft.com/sqlserver/en/us/common-criteria.aspx>

1.16.8 Audit Database Management Event Class (Level 1, Scorable)

Description:

Add the Audit Database Management Event Class to the audit trace.

Rationale:

This event class indicates that a database has been created, altered, or dropped..

Remediation:

Stop your trace if running and add the Audit Database Management Event Class to it:

```
-- Identify your trace by file name with the following query:
SELECT * FROM sys.fn_trace_getinfo(0) WHERE property = 2;

DECLARE @traceid int = <set your traceid here>;
--stop the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 0;
--add the new event class
EXECUTE sp_trace_setevent @TraceID, 128, 1, 1; -- TextData
EXECUTE sp_trace_setevent @TraceID, 128, 11, 1; -- LoginName
EXECUTE sp_trace_setevent @TraceID, 128, 14, 1; -- StartTime
EXECUTE sp_trace_setevent @TraceID, 128, 21, 1; -- EventSubClass
EXECUTE sp_trace_setevent @TraceID, 128, 23, 1; -- Success
EXECUTE sp_trace_setevent @TraceID, 128, 34, 1; -- ObjectName
EXECUTE sp_trace_setevent @TraceID, 128, 35, 1; -- DatabaseName
EXECUTE sp_trace_setevent @TraceID, 128, 40, 1; -- DBUserName
EXECUTE sp_trace_setevent @TraceID, 128, 64, 1; -- SessionLoginName
--start the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 1;
```

Audit:

Verify the events captured by the trace:

Run the T-SQL query below:

```
SELECT
    CAST(e.name as varchar(70)) AS Event_Descr,
    CAST(c.name as varchar(70)) AS Column_Descr
FROM sys.fn_trace_geteventinfo(1) t
    INNER JOIN sys.trace_events e
        ON t.eventID = e.trace_event_id
    INNER JOIN sys.trace_columns c
        ON t.columnid = c.trace_column_id
WHERE e.name = 'Audit Database Management Event'
```

```
ORDER BY e.name;
```

This query should return the columns listed in the Remediation section.

Default Value:

By default, this information is not captured in the default trace.

References:

<http://www.microsoft.com/sqlserver/en/us/common-criteria.aspx>

1.16.9 Audit Database Mirroring Login Event Class (Level 1, Scorable)

Description:

Add the Audit Database Mirroring Login Event Class to the audit trace.

Rationale:

This event class reports audit messages related to database mirroring transport security

Remediation:

Stop your trace if running and add the Audit Database Mirroring Login Event Class to it:

```
-- Identify your trace by file name with the following query:
SELECT * FROM sys.fn_trace_getinfo(0) WHERE property = 2;

DECLARE @traceid int = <set your traceid here>;
--stop the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 0;
--add the new event class
EXECUTE sp_trace_setevent @TraceID, 154, 1, 1; -- TextData
EXECUTE sp_trace_setevent @TraceID, 154, 11, 1; -- LoginName
EXECUTE sp_trace_setevent @TraceID, 154, 14, 1; -- StartTime
EXECUTE sp_trace_setevent @TraceID, 154, 21, 1; -- EventSubClass
EXECUTE sp_trace_setevent @TraceID, 154, 23, 1; -- Success
EXECUTE sp_trace_setevent @TraceID, 154, 34, 1; -- ObjectName
EXECUTE sp_trace_setevent @TraceID, 154, 35, 1; -- DatabaseName
EXECUTE sp_trace_setevent @TraceID, 154, 40, 1; -- DBUserName
EXECUTE sp_trace_setevent @TraceID, 154, 64, 1; -- SessionLoginName
--start the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 1;
```

Audit:

Verify the events captured by the trace:

Run the T-SQL query below:

```
SELECT
    CAST(e.name as varchar(70)) AS Event_Descr,
    CAST(c.name as varchar(70)) AS Column_Descr
FROM sys.fn_trace_geteventinfo(1) t
    INNER JOIN sys.trace_events e
        ON t.eventID = e.trace_event_id
    INNER JOIN sys.trace_columns c
        ON t.columnid = c.trace_column_id
WHERE e.name = 'Audit Database Mirroring Login Event'
```

```
ORDER BY e.name;
```

This query should return the columns listed in the Remediation section.

Default Value:

By default, this information is not captured in the default trace.

References:

<http://www.microsoft.com/sqlserver/en/us/common-criteria.aspx>

1.16.10 Audit Database Object Access Event Class (Level 2, Scorable)

Description:

Add the Audit Database Object Access Event Class to the audit trace.

Rationale:

This event class indicates that a database object, such as a schema, has been accessed.

Remediation:

Stop your trace if running and add the Audit Database Object Access Event Class to it:

```
-- Identify your trace by file name with the following query:
SELECT * FROM sys.fn_trace_getinfo(0) WHERE property = 2;

DECLARE @traceid int = <set your traceid here>;
--stop the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 0;
--add the new event class
EXECUTE sp_trace_setevent @TraceID, 180, 1, 1; -- TextData
EXECUTE sp_trace_setevent @TraceID, 180, 11, 1; -- LoginName
EXECUTE sp_trace_setevent @TraceID, 180, 14, 1; -- StartTime
EXECUTE sp_trace_setevent @TraceID, 180, 23, 1; -- Success
EXECUTE sp_trace_setevent @TraceID, 180, 34, 1; -- ObjectName
EXECUTE sp_trace_setevent @TraceID, 180, 35, 1; -- DatabaseName
EXECUTE sp_trace_setevent @TraceID, 180, 40, 1; -- DBUserName
EXECUTE sp_trace_setevent @TraceID, 180, 64, 1; -- SessionLoginName
--start the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 1;
```

Audit:

Verify the events captured by the trace:

Run the T-SQL query below:

```
SELECT
    CAST(e.name as varchar(70)) AS Event_Descr,
    CAST(c.name as varchar(70)) AS Column_Descr
FROM sys.fn_trace_geteventinfo(1) t
    INNER JOIN sys.trace_events e
        ON t.eventID = e.trace_event_id
    INNER JOIN sys.trace_columns c
        ON t.columnid = c.trace_column_id
WHERE e.name = 'Audit Database Object Access Event'
ORDER BY e.name;
```


This query should return the columns listed in the Remediation section.

Default Value:

By default, this information is not captured in the default trace.

References:

<http://www.microsoft.com/sqlserver/en/us/common-criteria.aspx>

1.16.11 Audit Database Object GDR Event Class (Level 1, Scorable)

Description:

Add the Audit Database Object GDR Event Class to the audit trace.

Rationale:

This event class indicates that a GDR event for a database object has occurred.

Remediation:

Stop your trace if running and add the Audit Database Object GDR Event Class to it:

```
-- Identify your trace by file name with the following query:
SELECT * FROM sys.fn_trace_getinfo(0) WHERE property = 2;

DECLARE @traceid int = <set your traceid here>;
--stop the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 0;
--add the new event class
EXECUTE sp_trace_setevent @TraceID, 172, 1, 1; -- TextData
EXECUTE sp_trace_setevent @TraceID, 172, 11, 1; -- LoginName
EXECUTE sp_trace_setevent @TraceID, 172, 14, 1; -- StartTime
EXECUTE sp_trace_setevent @TraceID, 172, 21, 1; -- EventSubClass
EXECUTE sp_trace_setevent @TraceID, 172, 23, 1; -- Success
EXECUTE sp_trace_setevent @TraceID, 172, 34, 1; -- ObjectName
EXECUTE sp_trace_setevent @TraceID, 172, 35, 1; -- DatabaseName
EXECUTE sp_trace_setevent @TraceID, 172, 40, 1; -- DBUserName
EXECUTE sp_trace_setevent @TraceID, 172, 64, 1; -- SessionLoginName
--start the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 1;
```

Audit:

Verify the events captured by the trace:

Run the T-SQL query below:

```
SELECT
    CAST(e.name as varchar(70)) AS Event_Descr,
    CAST(c.name as varchar(70)) AS Column_Descr
FROM sys.fn_trace_geteventinfo(1) t
    INNER JOIN sys.trace_events e
        ON t.eventID = e.trace_event_id
    INNER JOIN sys.trace_columns c
        ON t.columnid = c.trace_column_id
WHERE e.name = 'Audit Database Object GDR Event'
ORDER BY e.name;
```

This query should return the columns listed in the Remediation section.

Default Value:

By default, this information is not captured in the default trace.

References:

<http://www.microsoft.com/sqlserver/en/us/common-criteria.aspx>

1.16.12 Audit Database Object Management Event Class (Level 1, Scorable)

Description:

Add the Audit Database Object Management Event Class to the audit trace.

Rationale:

This event class indicates that a CREATE, ALTER, or DROP statement was executed on a database object.

Remediation:

Stop your trace if running and add the Audit Database Object Management Event Class to it:

```
-- Identify your trace by file name with the following query:
SELECT * FROM sys.fn_trace_getinfo(0) WHERE property = 2;

DECLARE @traceid int = <set your traceid here>;
--stop the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 0;
--add the new event class
EXECUTE sp_trace_setevent @TraceID, 129, 1, 1; -- TextData
EXECUTE sp_trace_setevent @TraceID, 129, 11, 1; -- LoginName
EXECUTE sp_trace_setevent @TraceID, 129, 14, 1; -- StartTime
EXECUTE sp_trace_setevent @TraceID, 129, 21, 1; -- EventSubClass
EXECUTE sp_trace_setevent @TraceID, 129, 23, 1; -- Success
EXECUTE sp_trace_setevent @TraceID, 129, 34, 1; -- ObjectName
EXECUTE sp_trace_setevent @TraceID, 129, 35, 1; -- DatabaseName
EXECUTE sp_trace_setevent @TraceID, 129, 40, 1; -- DBUserName
EXECUTE sp_trace_setevent @TraceID, 129, 64, 1; -- SessionLoginName
--start the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 1;
```

Audit:

Verify the events captured by the trace:

Run the T-SQL query below:

```
SELECT
    CAST(e.name as varchar(70)) AS Event_Descr,
    CAST(c.name as varchar(70)) AS Column_Descr
FROM sys.fn_trace_geteventinfo(1) t
    INNER JOIN sys.trace_events e
        ON t.eventID = e.trace_event_id
    INNER JOIN sys.trace_columns c
```

```

        ON t.columnid = c.trace_column_id
WHERE e.name = 'Audit Database Object Management Event'
ORDER BY e.name;

```

This query should return the columns listed in the Remediation section.

Default Value:

By default, this information is not captured in the default trace.

References:

<http://www.microsoft.com/sqlserver/en/us/common-criteria.aspx>

1.16.13 *Audit Database Object Take Ownership Event Class (Level 1, Scorable)*

Description:

Add the Audit Database Object Take Ownership Event Class to the audit trace.

Rationale:

This event class indicates that there has been a change of owner for objects in database scope.

Remediation:

Stop your trace if running and add the Audit Database Object Take Ownership Event Class to it:

```

-- Identify your trace by file name with the following query:
SELECT * FROM sys.fn_trace_getinfo(0) WHERE property = 2;

DECLARE @traceid int = <set your traceid here>;
--stop the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 0;
--add the new event class
EXECUTE sp_trace_setevent @TraceID, 135, 1, 1; -- TextData
EXECUTE sp_trace_setevent @TraceID, 135, 11, 1; -- LoginName
EXECUTE sp_trace_setevent @TraceID, 135, 14, 1; -- StartTime
EXECUTE sp_trace_setevent @TraceID, 135, 23, 1; -- Success
EXECUTE sp_trace_setevent @TraceID, 135, 34, 1; -- ObjectName
EXECUTE sp_trace_setevent @TraceID, 135, 35, 1; -- DatabaseName
EXECUTE sp_trace_setevent @TraceID, 135, 40, 1; -- DBUserName
EXECUTE sp_trace_setevent @TraceID, 135, 64, 1; -- SessionLoginName
--start the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 1;

```

Audit:

Verify the events captured by the trace:

Run the T-SQL query below:

```

SELECT
    CAST(e.name as varchar(70)) AS Event_Descr,
    CAST(c.name as varchar(70)) AS Column_Descr
FROM sys.fn_trace_geteventinfo(1) t

```

```

INNER JOIN sys.trace_events e
    ON t.eventID = e.trace_event_id
INNER JOIN sys.trace_columns c
    ON t.columnid = c.trace_column_id
WHERE e.name = 'Audit Database Object Take Ownership Event'
ORDER BY e.name;

```

This query should return the columns listed in the Remediation section.

Default Value:

By default, this information is not captured in the default trace.

References:

<http://www.microsoft.com/sqlserver/en/us/common-criteria.aspx>

1.16.14 *Audit Database Principal Impersonation Event Class (Level 1, Scorable)*

Description:

Add the Audit Database Principal Impersonation Event Class to the audit trace.

Rationale:

This event class indicates that an impersonation has occurred within the database scope.

Remediation:

Stop your trace if running and add the Audit Database Principal Impersonation Event Class to it:

```

-- Identify your trace by file name with the following query:
SELECT * FROM sys.fn_trace_getinfo(0) WHERE property = 2;

DECLARE @traceid int = <set your traceid here>;
--stop the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 0;
--add the new event class
EXECUTE sp_trace_setevent @TraceID, 133, 1, 1; -- TextData
EXECUTE sp_trace_setevent @TraceID, 133, 11, 1; -- LoginName
EXECUTE sp_trace_setevent @TraceID, 133, 14, 1; -- StartTime
EXECUTE sp_trace_setevent @TraceID, 133, 21, 1; -- EventSubClass
EXECUTE sp_trace_setevent @TraceID, 133, 23, 1; -- Success
EXECUTE sp_trace_setevent @TraceID, 133, 34, 1; -- ObjectName
EXECUTE sp_trace_setevent @TraceID, 133, 35, 1; -- DatabaseName
EXECUTE sp_trace_setevent @TraceID, 133, 40, 1; -- DBUserName
EXECUTE sp_trace_setevent @TraceID, 133, 64, 1; -- SessionLoginName
--start the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 1;

```

Audit:

Verify the events captured by the trace:

Run the T-SQL query below:

```

SELECT

```

```

        CAST(e.name as varchar(70)) AS Event_Descr,
        CAST(c.name as varchar(70)) AS Column_Descr
FROM sys.fn_trace_geteventinfo(1) t
    INNER JOIN sys.trace_events e
        ON t.eventID = e.trace_event_id
    INNER JOIN sys.trace_columns c
        ON t.columnid = c.trace_column_id
WHERE e.name = 'Audit Database Principal Impersonation Event'
ORDER BY e.name;

```

This query should return the columns listed in the Remediation section.

Default Value:

By default, this information is not captured in the default trace.

References:

<http://www.microsoft.com/sqlserver/en/us/common-criteria.aspx>

1.16.15 Audit Database Principal Management Event Class (Level 1, Scorable)

Description:

Add the Audit Database Principal Management Event Class to the audit trace.

Rationale:

This event class indicates that principals have been created, altered, or dropped from a database.

Remediation:

Stop your trace if running and add the Audit Database Principal Management Event Class to it:

```

-- Identify your trace by file name with the following query:
SELECT * FROM sys.fn_trace_getinfo(0) WHERE property = 2;

DECLARE @traceid int = <set your traceid here>;
--stop the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 0;
--add the new event class
EXECUTE sp_trace_setevent @TraceID, 130, 1, 1; -- TextData
EXECUTE sp_trace_setevent @TraceID, 130, 11, 1; -- LoginName
EXECUTE sp_trace_setevent @TraceID, 130, 14, 1; -- StartTime
EXECUTE sp_trace_setevent @TraceID, 130, 21, 1; -- EventSubClass
EXECUTE sp_trace_setevent @TraceID, 130, 23, 1; -- Success
EXECUTE sp_trace_setevent @TraceID, 130, 34, 1; -- ObjectName
EXECUTE sp_trace_setevent @TraceID, 130, 35, 1; -- DatabaseName
EXECUTE sp_trace_setevent @TraceID, 130, 40, 1; -- DBUserName
EXECUTE sp_trace_setevent @TraceID, 130, 64, 1; -- SessionLoginName
--start the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 1;

```

Audit:

Verify the events captured by the trace:

Run the T-SQL query below:

```
SELECT
    CAST(e.name as varchar(70)) AS Event_Descr,
    CAST(c.name as varchar(70)) AS Column_Descr
FROM sys.fn_trace_geteventinfo(1) t
    INNER JOIN sys.trace_events e
        ON t.eventID = e.trace_event_id
    INNER JOIN sys.trace_columns c
        ON t.columnid = c.trace_column_id
WHERE e.name = 'Audit Database Principal Management Event'
ORDER BY e.name;
```

This query should return the columns listed in the Remediation section.

Default Value:

By default, this information is not captured in the default trace.

References:

<http://www.microsoft.com/sqlserver/en/us/common-criteria.aspx>

1.16.16 Audit Database Scope GDR Event Class (Level 1, Scorable)

Description:

Add the Audit Database Scope GDR Event Class to the audit trace.

Rationale:

This event class indicates that a GRANT, REVOKE, or DENY has been issued for a statement permission by a user in Microsoft SQL Server.

Remediation:

Stop your trace if running and add the Audit Database Scope GDR Event Class to it:

```
-- Identify your trace by file name with the following query:
SELECT * FROM sys.fn_trace_getinfo(0) WHERE property = 2;

DECLARE @traceid int = <set your traceid here>;
--stop the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 0;
--add the new event class
EXECUTE sp_trace_setevent @TraceID, 102, 1, 1; -- TextData
EXECUTE sp_trace_setevent @TraceID, 102, 11, 1; -- LoginName
EXECUTE sp_trace_setevent @TraceID, 102, 14, 1; -- StartTime
EXECUTE sp_trace_setevent @TraceID, 102, 21, 1; -- EventSubClass
EXECUTE sp_trace_setevent @TraceID, 102, 23, 1; -- Success
EXECUTE sp_trace_setevent @TraceID, 102, 34, 1; -- ObjectName
EXECUTE sp_trace_setevent @TraceID, 102, 35, 1; -- DatabaseName
EXECUTE sp_trace_setevent @TraceID, 102, 40, 1; -- DBUserName
EXECUTE sp_trace_setevent @TraceID, 102, 64, 1; -- SessionLoginName
--start the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 1;
```

Audit:

Verify the events captured by the trace:

Run the T-SQL query below:

```
SELECT
    CAST(e.name as varchar(70)) AS Event_Descr,
    CAST(c.name as varchar(70)) AS Column_Descr
FROM sys.fn_trace_geteventinfo(1) t
    INNER JOIN sys.trace_events e
        ON t.eventID = e.trace_event_id
    INNER JOIN sys.trace_columns c
        ON t.columnid = c.trace_column_id
WHERE e.name = 'Audit Database Scope GDR Event'
ORDER BY e.name;
```

This query should return the columns listed in the Remediation section.

Default Value:

By default, this information is captured in the default trace.

References:

<http://www.microsoft.com/sqlserver/en/us/common-criteria.aspx>

1.16.17 Audit DBCC Event Class (Level 1, Scorable)

Description:

Add the Audit DBCC Event Class to the audit trace.

Rationale:

This event class indicates that a DBCC command has been issued.

Remediation:

Stop your trace if running and add the Audit DBCC Event Class to it:

```
-- Identify your trace by file name with the following query:
SELECT * FROM sys.fn_trace_getinfo(0) WHERE property = 2;

DECLARE @traceid int = <set your traceid here>;
--stop the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 0;
--add the new event class
EXECUTE sp_trace_setevent @TraceID, 116, 1, 1; -- TextData
EXECUTE sp_trace_setevent @TraceID, 116, 11, 1; -- LoginName
EXECUTE sp_trace_setevent @TraceID, 116, 14, 1; -- StartTime
EXECUTE sp_trace_setevent @TraceID, 116, 23, 1; -- Success
EXECUTE sp_trace_setevent @TraceID, 116, 35, 1; -- DatabaseName
EXECUTE sp_trace_setevent @TraceID, 116, 40, 1; -- DBUserName
EXECUTE sp_trace_setevent @TraceID, 116, 64, 1; -- SessionLoginName
--start the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 1;
```

Audit:

Verify the events captured by the trace:

Run the T-SQL query below:

```
SELECT
    CAST(e.name as varchar(70)) AS Event_Descr,
    CAST(c.name as varchar(70)) AS Column_Descr
FROM sys.fn_trace_geteventinfo(1) t
    INNER JOIN sys.trace_events e
        ON t.eventID = e.trace_event_id
    INNER JOIN sys.trace_columns c
        ON t.columnid = c.trace_column_id
WHERE e.name = 'Audit DBCC Event'
ORDER BY e.name;
```

This query should return the columns listed in the Remediation section.

Default Value:

By default, this information is captured in the default trace.

References:

<http://www.microsoft.com/sqlserver/en/us/common-criteria.aspx>

1.16.18 Audit Login Change Password Event Class (Level 1, Scorable)

Description:

Add the Audit Login Change Password Event Class to the audit trace.

Rationale:

This event class indicates that a user has changed their SQL Server login password.

Remediation:

Stop your trace if running and add the Audit Login Change Password Event Class to it:

```
-- Identify your trace by file name with the following query:
SELECT * FROM sys.fn_trace_getinfo(0) WHERE property = 2;

DECLARE @traceid int = <set your traceid here>;
--stop the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 0;
--add the new event class
EXECUTE sp_trace_setevent @TraceID, 107, 1, 1; -- TextData
EXECUTE sp_trace_setevent @TraceID, 107, 11, 1; -- LoginName
EXECUTE sp_trace_setevent @TraceID, 107, 14, 1; -- StartTime
EXECUTE sp_trace_setevent @TraceID, 107, 21, 1; -- EventSubClass
EXECUTE sp_trace_setevent @TraceID, 107, 23, 1; -- Success
EXECUTE sp_trace_setevent @TraceID, 107, 34, 1; -- ObjectName
EXECUTE sp_trace_setevent @TraceID, 107, 35, 1; -- DatabaseName
EXECUTE sp_trace_setevent @TraceID, 107, 40, 1; -- DBUserName
EXECUTE sp_trace_setevent @TraceID, 107, 64, 1; -- SessionLoginName
--start the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 1;
```

Audit:

Verify the events captured by the trace:

Run the T-SQL query below:

```
SELECT
    CAST(e.name as varchar(70)) AS Event_Descr,
    CAST(c.name as varchar(70)) AS Column_Descr
FROM sys.fn_trace_geteventinfo(1) t
    INNER JOIN sys.trace_events e
        ON t.eventID = e.trace_event_id
    INNER JOIN sys.trace_columns c
        ON t.columnid = c.trace_column_id
WHERE e.name = 'Audit Login Change Password Event'
ORDER BY e.name;
```

This query should return the columns listed in the Remediation section.

Default Value:

By default, this information is not captured in the default trace.

References:

<http://www.microsoft.com/sqlserver/en/us/common-criteria.aspx>

1.16.19 Audit Login Change Property Event Class (Level 1, Scorable)

Description:

Add the Audit Login Change Property Event Class to the audit trace.

Rationale:

This event class indicates that sp_defaultdb, sp_defaultlanguage, or ALTER LOGIN was used to modify a property of a login.

Remediation:

Stop your trace if running and add the Audit Login Change Property Event Class to it:

```
-- Identify your trace by file name with the following query:
SELECT * FROM sys.fn_trace_getinfo(0) WHERE property = 2;

DECLARE @traceid int = <set your traceid here>;
--stop the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 0;
--add the new event class
EXECUTE sp_trace_setevent @TraceID, 106, 1, 1; -- TextData
EXECUTE sp_trace_setevent @TraceID, 106, 11, 1; -- LoginName
EXECUTE sp_trace_setevent @TraceID, 106, 14, 1; -- StartTime
EXECUTE sp_trace_setevent @TraceID, 106, 21, 1; -- EventSubClass
EXECUTE sp_trace_setevent @TraceID, 106, 23, 1; -- Success
EXECUTE sp_trace_setevent @TraceID, 106, 34, 1; -- ObjectName
EXECUTE sp_trace_setevent @TraceID, 106, 35, 1; -- DatabaseName
EXECUTE sp_trace_setevent @TraceID, 106, 40, 1; -- DBUserName
EXECUTE sp_trace_setevent @TraceID, 106, 64, 1; -- SessionLoginName
--start the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 1;
```

Audit:

Verify the events captured by the trace:

Run the T-SQL query below:

```
SELECT
    CAST(e.name as varchar(70)) AS Event_Descr,
    CAST(c.name as varchar(70)) AS Column_Descr
FROM sys.fn_trace_geteventinfo(1) t
    INNER JOIN sys.trace_events e
        ON t.eventID = e.trace_event_id
    INNER JOIN sys.trace_columns c
        ON t.columnid = c.trace_column_id
WHERE e.name = 'Audit Login Change Property Event'
ORDER BY e.name;
```

This query should return the columns listed in the Remediation section.

Default Value:

By default, this information is captured in the default trace.

References:

<http://www.microsoft.com/sqlserver/en/us/common-criteria.aspx>

1.16.20 Audit Login Event Class (Level 1, Scorable)

Description:

Add the Audit Login Event Class to the audit trace.

Rationale:

This event class indicates that a user has successfully logged into SQL Server.

Remediation:

Stop your trace if running and add the Audit Login Event Class to it:

```
-- Identify your trace by file name with the following query:
SELECT * FROM sys.fn_trace_getinfo(0) WHERE property = 2;

DECLARE @traceid int = <set your traceid here>;
--stop the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 0;
--add the new event class
EXECUTE sp_trace_setevent @TraceID, 14, 1, 1; -- TextData
EXECUTE sp_trace_setevent @TraceID, 14, 11, 1; -- LoginName
EXECUTE sp_trace_setevent @TraceID, 14, 14, 1; -- StartTime
EXECUTE sp_trace_setevent @TraceID, 14, 21, 1; -- EventSubClass
EXECUTE sp_trace_setevent @TraceID, 14, 23, 1; -- Success
EXECUTE sp_trace_setevent @TraceID, 14, 64, 1; -- SessionLoginName
--start the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 1;
```

Audit:

Verify the events captured by the trace:

Run the T-SQL query below:

```
SELECT
    CAST(e.name as varchar(70)) AS Event_Descr,
    CAST(c.name as varchar(70)) AS Column_Descr
FROM sys.fn_trace_geteventinfo(1) t
    INNER JOIN sys.trace_events e
        ON t.eventID = e.trace_event_id
    INNER JOIN sys.trace_columns c
        ON t.columnid = c.trace_column_id
WHERE e.name = 'Audit Login Event'
ORDER BY e.name;
```

This query should return the columns listed in the Remediation section.

Default Value:

By default, this information is not captured in the default trace.

References:

<http://www.microsoft.com/sqlserver/en/us/common-criteria.aspx>

1.16.21 Audit Login Failed Event Class (Level 1, Scorable)

Description:

Add the Audit Login Failed Event Class to the audit trace.

Rationale:

This event class indicates that a user attempted to log in to SQL Server and failed.

Remediation:

Stop your trace if running and add the Audit Login Failed Event Class to it:

```
-- Identify your trace by file name with the following query:
SELECT * FROM sys.fn_trace_getinfo(0) WHERE property = 2;

DECLARE @traceid int = <set your traceid here>;
--stop the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 0;
--add the new event class
EXECUTE sp_trace_setevent @TraceID, 20, 1, 1; -- TextData
EXECUTE sp_trace_setevent @TraceID, 20, 11, 1; -- LoginName
EXECUTE sp_trace_setevent @TraceID, 20, 14, 1; -- StartTime
EXECUTE sp_trace_setevent @TraceID, 20, 21, 1; -- EventSubClass
EXECUTE sp_trace_setevent @TraceID, 20, 23, 1; -- Success
EXECUTE sp_trace_setevent @TraceID, 20, 31, 1; -- Error
EXECUTE sp_trace_setevent @TraceID, 20, 35, 1; -- DatabaseName
EXECUTE sp_trace_setevent @TraceID, 20, 64, 1; -- SessionLoginName
--start the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 1;
```

Audit:

Verify the events captured by the trace:

Run the T-SQL query below:

```

SELECT
    CAST(e.name as varchar(70)) AS Event_Descr,
    CAST(c.name as varchar(70)) AS Column_Descr
FROM sys.fn_trace_geteventinfo(1) t
    INNER JOIN sys.trace_events e
        ON t.eventID = e.trace_event_id
    INNER JOIN sys.trace_columns c
        ON t.columnid = c.trace_column_id
WHERE e.name = 'Audit Login Failed Event'
ORDER BY e.name;

```

This query should return the columns listed in the Remediation section.

Default Value:

By default, this information is captured in the default trace.

References:

<http://www.microsoft.com/sqlserver/en/us/common-criteria.aspx>

1.16.22 Audit Logout Event Class (Level 1, Scorable)

Description:

Add the Audit Logout Event Class to the audit trace.

Rationale:

This event class indicates that a user has logged out of SQL Server.

Remediation:

Stop your trace if running and add the Audit Logout Event Class to it:

```

-- Identify your trace by file name with the following query:
SELECT * FROM sys.fn_trace_getinfo(0) WHERE property = 2;

DECLARE @traceid int = <set your traceid here>;
--stop the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 0;
--add the new event class
EXECUTE sp_trace_setevent @TraceID, 15, 11, 1; -- LoginName
EXECUTE sp_trace_setevent @TraceID, 15, 14, 1; -- StartTime
EXECUTE sp_trace_setevent @TraceID, 15, 21, 1; -- EventSubClass
EXECUTE sp_trace_setevent @TraceID, 15, 23, 1; -- Success
EXECUTE sp_trace_setevent @TraceID, 15, 64, 1; -- SessionLoginName
--start the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 1;

```

Audit:

Verify the events captured by the trace:

Run the T-SQL query below:

```

SELECT
    CAST(e.name as varchar(70)) AS Event_Descr,
    CAST(c.name as varchar(70)) AS Column_Descr
FROM sys.fn_trace_geteventinfo(1) t

```

```

INNER JOIN sys.trace_events e
    ON t.eventID = e.trace_event_id
INNER JOIN sys.trace_columns c
    ON t.columnid = c.trace_column_id
WHERE e.name = 'Audit Logout Event'
ORDER BY e.name;

```

This query should return the columns listed in the Remediation section.

Default Value:

By default, this information is not captured in the default trace.

References:

<http://www.microsoft.com/sqlserver/en/us/common-criteria.aspx>

1.16.23 Audit Schema Object Access Event Class (Level 2, Scorable)

Description:

Add the Audit Schema Object Access Event Class to the audit trace.

Rationale:

This event class indicates that an object permission (such as SELECT) has been used.

Remediation:

Stop your trace if running and add the Audit Schema Object Access Event Class to it:

```

-- Identify your trace by file name with the following query:
SELECT * FROM sys.fn_trace_getinfo(0) WHERE property = 2;

DECLARE @traceid int = <set your traceid here>;
--stop the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 0;
--add the new event class
EXECUTE sp_trace_setevent @TraceID, 114, 1, 1; -- TextData
EXECUTE sp_trace_setevent @TraceID, 114, 11, 1; -- LoginName
EXECUTE sp_trace_setevent @TraceID, 114, 14, 1; -- StartTime
EXECUTE sp_trace_setevent @TraceID, 114, 21, 1; -- EventSubClass
EXECUTE sp_trace_setevent @TraceID, 114, 23, 1; -- Success
EXECUTE sp_trace_setevent @TraceID, 114, 34, 1; -- ObjectName
EXECUTE sp_trace_setevent @TraceID, 114, 35, 1; -- DatabaseName
EXECUTE sp_trace_setevent @TraceID, 114, 40, 1; -- DBUserName
EXECUTE sp_trace_setevent @TraceID, 114, 59, 1; -- ParentName
EXECUTE sp_trace_setevent @TraceID, 114, 64, 1; -- SessionLoginName
--start the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 1;

```

Audit:

Verify the events captured by the trace:

Run the T-SQL query below:

```

SELECT
    CAST(e.name as varchar(70)) AS Event_Descr,
    CAST(c.name as varchar(70)) AS Column_Descr
FROM sys.fn_trace_geteventinfo(1) t

```

```

INNER JOIN sys.trace_events e
    ON t.eventID = e.trace_event_id
INNER JOIN sys.trace_columns c
    ON t.columnid = c.trace_column_id
WHERE e.name = 'Audit Schema Object Access Event'
ORDER BY e.name;

```

This query should return the columns listed in the Remediation section.

Default Value:

By default, this information is captured in the default trace.

References:

<http://www.microsoft.com/sqlserver/en/us/common-criteria.aspx>

1.16.24 Audit Schema Object GDR Event Class (Level 1, Scorable)

Description:

Add the Audit Schema Object GDR Event Class to the audit trace.

Rationale:

This event class indicates that a GRANT, REVOKE, or DENY was issued for a schema object permission by a user in SQL Server.

Remediation:

Stop your trace if running and add the Audit Schema Object GDR Event Class to it:

```

-- Identify your trace by file name with the following query:
SELECT * FROM sys.fn_trace_getinfo(0) WHERE property = 2;

DECLARE @traceid int = <set your traceid here>;
--stop the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 0;
--add the new event class
EXECUTE sp_trace_setevent @TraceID, 103, 1, 1; -- TextData
EXECUTE sp_trace_setevent @TraceID, 103, 11, 1; -- LoginName
EXECUTE sp_trace_setevent @TraceID, 103, 14, 1; -- StartTime
EXECUTE sp_trace_setevent @TraceID, 103, 21, 1; -- EventSubClass
EXECUTE sp_trace_setevent @TraceID, 103, 23, 1; -- Success
EXECUTE sp_trace_setevent @TraceID, 103, 34, 1; -- ObjectName
EXECUTE sp_trace_setevent @TraceID, 103, 35, 1; -- DatabaseName
EXECUTE sp_trace_setevent @TraceID, 103, 40, 1; -- DBUserName
EXECUTE sp_trace_setevent @TraceID, 103, 59, 1; -- ParentName
EXECUTE sp_trace_setevent @TraceID, 103, 64, 1; -- SessionLoginName
--start the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 1;

```

Audit:

Verify the events captured by the trace:

Run the T-SQL query below:

```

SELECT
    CAST(e.name as varchar(70)) AS Event_Descr,

```

```

        CAST(c.name as varchar(70)) AS Column_Descr
FROM sys.fn_trace_geteventinfo(1) t
    INNER JOIN sys.trace_events e
        ON t.eventID = e.trace_event_id
    INNER JOIN sys.trace_columns c
        ON t.columnid = c.trace_column_id
WHERE e.name = 'Audit Schema Object GDR Event'
ORDER BY e.name;

```

This query should return the columns listed in the Remediation section.

Default Value:

By default, this information is captured in the default trace.

References:

<http://www.microsoft.com/sqlserver/en/us/common-criteria.aspx>

1.16.25 Audit Schema Object Take Ownership Event Class (Level 1, Scorable)

Description:

Add the Audit Schema Object Take Ownership Event Class to the audit trace.

Rationale:

This event class indicates that the permissions to change the owner of schema object have been checked.

Remediation:

Stop your trace if running and add the Audit Schema Object Take Ownership Event Class to it:

```

-- Identify your trace by file name with the following query:
SELECT * FROM sys.fn_trace_getinfo(0) WHERE property = 2;

DECLARE @traceid int = <set your traceid here>;
--stop the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 0;
--add the new event class
EXECUTE sp_trace_setevent @TraceID, 153, 1, 1; -- TextData
EXECUTE sp_trace_setevent @TraceID, 153, 11, 1; -- LoginName
EXECUTE sp_trace_setevent @TraceID, 153, 14, 1; -- StartTime
EXECUTE sp_trace_setevent @TraceID, 153, 23, 1; -- Success
EXECUTE sp_trace_setevent @TraceID, 153, 34, 1; -- ObjectName
EXECUTE sp_trace_setevent @TraceID, 153, 35, 1; -- DatabaseName
EXECUTE sp_trace_setevent @TraceID, 153, 40, 1; -- DBUserName
EXECUTE sp_trace_setevent @TraceID, 153, 59, 1; -- ParentName
EXECUTE sp_trace_setevent @TraceID, 153, 64, 1; -- SessionLoginName
--start the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 1;

```

Audit:

Verify the events captured by the trace:

Run the T-SQL query below:

```
SELECT
    CAST(e.name as varchar(70)) AS Event_Descr,
    CAST(c.name as varchar(70)) AS Column_Descr
FROM sys.fn_trace_geteventinfo(1) t
    INNER JOIN sys.trace_events e
        ON t.eventID = e.trace_event_id
    INNER JOIN sys.trace_columns c
        ON t.columnid = c.trace_column_id
WHERE e.name = 'Audit Schema Object Take Ownership Event'
ORDER BY e.name;
```

This query should return the columns listed in the Remediation section.

Default Value:

By default, this information is captured in the default trace.

References:

<http://www.microsoft.com/sqlserver/en/us/common-criteria.aspx>

1.16.26 Audit Server Alter Trace Event Class (Level 1, Scorable)

Description:

Add the Audit Server Alter Trace Event Class to the audit trace.

Rationale:

This event class indicates that the ALTER TRACE permission has been checked.

Remediation:

Stop your trace if running and add the Audit Server Alter Trace Event Class to it:

```
-- Identify your trace by file name with the following query:
SELECT * FROM sys.fn_trace_getinfo(0) WHERE property = 2;

DECLARE @traceid int = <set your traceid here>;
--stop the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 0;
--add the new event class
EXECUTE sp_trace_setevent @TraceID, 175, 1, 1; -- TextData
EXECUTE sp_trace_setevent @TraceID, 175, 11, 1; -- LoginName
EXECUTE sp_trace_setevent @TraceID, 175, 14, 1; -- StartTime
EXECUTE sp_trace_setevent @TraceID, 175, 23, 1; -- Success
EXECUTE sp_trace_setevent @TraceID, 175, 34, 1; -- ObjectName
EXECUTE sp_trace_setevent @TraceID, 175, 35, 1; -- DatabaseName
EXECUTE sp_trace_setevent @TraceID, 175, 40, 1; -- DBUserName
EXECUTE sp_trace_setevent @TraceID, 175, 64, 1; -- SessionLoginName
--start the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 1;
```

Audit:

Verify the events captured by the trace:

Run the T-SQL query below:

```
SELECT
    CAST(e.name as varchar(70)) AS Event_Descr,
    CAST(c.name as varchar(70)) AS Column_Descr
FROM sys.fn_trace_geteventinfo(1) t
    INNER JOIN sys.trace_events e
        ON t.eventID = e.trace_event_id
    INNER JOIN sys.trace_columns c
        ON t.columnid = c.trace_column_id
WHERE e.name = 'Audit Server Alter Trace Event'
ORDER BY e.name;
```

This query should return the columns listed in the Remediation section.

Default Value:

By default, this information is captured in the default trace.

References:

<http://www.microsoft.com/sqlserver/en/us/common-criteria.aspx>

1.16.27 Audit Server Object GDR Event Class (Level 1, Scorable)

Description:

Add the Audit Server Object GDR Event Class to the audit trace.

Rationale:

This event class indicates that a GDR event for a schema object has occurred.

Remediation:

Stop your trace if running and add the Audit Server Object GDR Event Class to it:

```
-- Identify your trace by file name with the following query:
SELECT * FROM sys.fn_trace_getinfo(0) WHERE property = 2;

DECLARE @traceid int = <set your traceid here>;
--stop the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 0;
--add the new event class
EXECUTE sp_trace_setevent @TraceID, 171, 1, 1; -- TextData
EXECUTE sp_trace_setevent @TraceID, 171, 11, 1; -- LoginName
EXECUTE sp_trace_setevent @TraceID, 171, 14, 1; -- StartTime
EXECUTE sp_trace_setevent @TraceID, 171, 21, 1; -- EventSubClass
EXECUTE sp_trace_setevent @TraceID, 171, 23, 1; -- Success
EXECUTE sp_trace_setevent @TraceID, 171, 34, 1; -- ObjectName
EXECUTE sp_trace_setevent @TraceID, 171, 35, 1; -- DatabaseName
EXECUTE sp_trace_setevent @TraceID, 171, 40, 1; -- DBUserName
EXECUTE sp_trace_setevent @TraceID, 171, 64, 1; -- SessionLoginName
--start the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 1;
```

Audit:

Verify the events captured by the trace:

Run the T-SQL query below:

```
SELECT
    CAST(e.name as varchar(70)) AS Event_Descr,
    CAST(c.name as varchar(70)) AS Column_Descr
FROM sys.fn_trace_geteventinfo(1) t
    INNER JOIN sys.trace_events e
        ON t.eventID = e.trace_event_id
    INNER JOIN sys.trace_columns c
        ON t.columnid = c.trace_column_id
WHERE e.name = 'Audit Server Object GDR Event'
ORDER BY e.name;
```

This query should return the columns listed in the Remediation section.

Default Value:

By default, this information is not captured in the default trace.

References:

<http://www.microsoft.com/sqlserver/en/us/common-criteria.aspx>

1.16.28 Audit Server Object Management Event Class (Level 1, Scorable)

Description:

Add the Audit Server Object Management Event Class to the audit trace.

Rationale:

This event class indicates that a CREATE, ALTER, or DROP event has occurred for a server object.

Remediation:

Stop your trace if running and add the Audit Server Object Management Event Class to it:

```
-- Identify your trace by file name with the following query:
SELECT * FROM sys.fn_trace_getinfo(0) WHERE property = 2;

DECLARE @traceid int = <set your traceid here>;
--stop the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 0;
--add the new event class
EXECUTE sp_trace_setevent @TraceID, 176, 1, 1; -- TextData
EXECUTE sp_trace_setevent @TraceID, 176, 11, 1; -- LoginName
EXECUTE sp_trace_setevent @TraceID, 176, 14, 1; -- StartTime
EXECUTE sp_trace_setevent @TraceID, 176, 21, 1; -- EventSubClass
EXECUTE sp_trace_setevent @TraceID, 176, 23, 1; -- Success
EXECUTE sp_trace_setevent @TraceID, 176, 34, 1; -- ObjectName
EXECUTE sp_trace_setevent @TraceID, 176, 35, 1; -- DatabaseName
EXECUTE sp_trace_setevent @TraceID, 176, 40, 1; -- DBUserName
EXECUTE sp_trace_setevent @TraceID, 176, 64, 1; -- SessionLoginName
--start the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 1;
```

Audit:

Verify the events captured by the trace:

Run the T-SQL query below:

```
SELECT
    CAST(e.name as varchar(70)) AS Event_Descr,
    CAST(c.name as varchar(70)) AS Column_Descr
FROM sys.fn_trace_geteventinfo(1) t
    INNER JOIN sys.trace_events e
        ON t.eventID = e.trace_event_id
    INNER JOIN sys.trace_columns c
        ON t.columnid = c.trace_column_id
WHERE e.name = 'Audit Server Object Management Event'
ORDER BY e.name;
```

This query should return the columns listed in the Remediation section.

Default Value:

By default, this information is not captured in the default trace.

References:

<http://www.microsoft.com/sqlserver/en/us/common-criteria.aspx>

1.16.29 Audit Server Object Take Ownership Event Class (Level 1, Scorable)

Description:

Add the Audit Server Object Take Ownership Event Class to the audit trace.

Rationale:

This event class indicates that a server object owner has changed.

Remediation:

Stop your trace if running and add the Audit Server Object Take Ownership Event Class to it:

```
-- Identify your trace by file name with the following query:
SELECT * FROM sys.fn_trace_getinfo(0) WHERE property = 2;

DECLARE @traceid int = <set your traceid here>;
--stop the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 0;
--add the new event class
EXECUTE sp_trace_setevent @TraceID, 134, 1, 1; -- TextData
EXECUTE sp_trace_setevent @TraceID, 134, 11, 1; -- LoginName
EXECUTE sp_trace_setevent @TraceID, 134, 14, 1; -- StartTime
EXECUTE sp_trace_setevent @TraceID, 134, 23, 1; -- Success
EXECUTE sp_trace_setevent @TraceID, 134, 34, 1; -- ObjectName
EXECUTE sp_trace_setevent @TraceID, 134, 35, 1; -- DatabaseName
EXECUTE sp_trace_setevent @TraceID, 134, 40, 1; -- DBUserName
EXECUTE sp_trace_setevent @TraceID, 134, 64, 1; -- SessionLoginName
--start the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 1;
```

Audit:

Verify the events captured by the trace:

Run the T-SQL query below:

```
SELECT
    CAST(e.name as varchar(70)) AS Event_Descr,
    CAST(c.name as varchar(70)) AS Column_Descr
FROM sys.fn_trace_geteventinfo(1) t
    INNER JOIN sys.trace_events e
        ON t.eventID = e.trace_event_id
    INNER JOIN sys.trace_columns c
        ON t.columnid = c.trace_column_id
WHERE e.name = 'Audit Server Object Take Ownership Event'
ORDER BY e.name;
```

This query should return the columns listed in the Remediation section.

Default Value:

By default, this information is not captured in the default trace.

References:

<http://www.microsoft.com/sqlserver/en/us/common-criteria.aspx>

1.16.30 Audit Server Operation Event Class (Level 1, Scorable)

Description:

Add the Audit Server Operation Event Class to the audit trace.

Rationale:

This event class indicates that Audit operations have occurred in the server.

Remediation:

Stop your trace if running and add the Audit Server Operation Event Class to it:

```
-- Identify your trace by file name with the following query:
SELECT * FROM sys.fn_trace_getinfo(0) WHERE property = 2;

DECLARE @traceid int = <set your traceid here>;
--stop the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 0;
--add the new event class
EXECUTE sp_trace_setevent @TraceID, 173, 1, 1; -- TextData
EXECUTE sp_trace_setevent @TraceID, 173, 11, 1; -- LoginName
EXECUTE sp_trace_setevent @TraceID, 173, 14, 1; -- StartTime
EXECUTE sp_trace_setevent @TraceID, 173, 21, 1; -- EventSubClass
EXECUTE sp_trace_setevent @TraceID, 173, 23, 1; -- Success
EXECUTE sp_trace_setevent @TraceID, 173, 34, 1; -- ObjectName
EXECUTE sp_trace_setevent @TraceID, 173, 35, 1; -- DatabaseName
EXECUTE sp_trace_setevent @TraceID, 173, 40, 1; -- DBUserName
EXECUTE sp_trace_setevent @TraceID, 173, 64, 1; -- SessionLoginName
--start the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 1;
```

Audit:

Verify the events captured by the trace:

Run the T-SQL query below:

```
SELECT
    CAST(e.name as varchar(70)) AS Event_Descr,
    CAST(c.name as varchar(70)) AS Column_Descr
FROM sys.fn_trace_geteventinfo(1) t
    INNER JOIN sys.trace_events e
        ON t.eventID = e.trace_event_id
    INNER JOIN sys.trace_columns c
        ON t.columnid = c.trace_column_id
WHERE e.name = 'Audit Server Operation Event'
ORDER BY e.name;
```

This query should return the columns listed in the Remediation section.

Default Value:

By default, this information is not captured in the default trace.

References:

<http://www.microsoft.com/sqlserver/en/us/common-criteria.aspx>

1.16.31 Audit Server Principal Impersonation Event Class (Level 1, Scorable)

Description:

Add the Audit Server Principal Impersonation Event Class to the audit trace.

Rationale:

This event class indicates that an impersonation has occurred within the server scope.

Remediation:

Stop your trace if running and add the Audit Server Principal Impersonation Event Class to it:

```
-- Identify your trace by file name with the following query:
SELECT * FROM sys.fn_trace_getinfo(0) WHERE property = 2;

DECLARE @traceid int = <set your traceid here>;
--stop the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 0;
--add the new event class
EXECUTE sp_trace_setevent @TraceID, 132, 1, 1; -- TextData
EXECUTE sp_trace_setevent @TraceID, 132, 11, 1; -- LoginName
EXECUTE sp_trace_setevent @TraceID, 132, 14, 1; -- StartTime
EXECUTE sp_trace_setevent @TraceID, 132, 21, 1; -- EventSubClass
EXECUTE sp_trace_setevent @TraceID, 132, 23, 1; -- Success
EXECUTE sp_trace_setevent @TraceID, 132, 34, 1; -- ObjectName
EXECUTE sp_trace_setevent @TraceID, 132, 35, 1; -- DatabaseName
EXECUTE sp_trace_setevent @TraceID, 132, 40, 1; -- DBUserName
```

```
EXECUTE sp_trace_setevent @TraceID, 132, 64, 1; -- SessionLoginName
--start the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 1;
```

Audit:

Verify the events captured by the trace:

Run the T-SQL query below:

```
SELECT
    CAST(e.name as varchar(70)) AS Event_Descr,
    CAST(c.name as varchar(70)) AS Column_Descr
FROM sys.fn_trace_geteventinfo(1) t
    INNER JOIN sys.trace_events e
        ON t.eventID = e.trace_event_id
    INNER JOIN sys.trace_columns c
        ON t.columnid = c.trace_column_id
WHERE e.name = 'Audit Server Principal Impersonation Event'
ORDER BY e.name;
```

This query should return the columns listed in the Remediation section.

Default Value:

By default, this information is not captured in the default trace.

References:

<http://www.microsoft.com/sqlserver/en/us/common-criteria.aspx>

1.16.32 Audit Server Principal Management Event Class (Level 1, Scorable)

Description:

Add the Audit Server Principal Management Event Class to the audit trace.

Rationale:

This event class indicates that a CREATE, ALTER, or DROP has occurred for a server principal.

Remediation:

Stop your trace if running and add the Audit Server Principal Management Event Class to it:

```
-- Identify your trace by file name with the following query:
SELECT * FROM sys.fn_trace_getinfo(0) WHERE property = 2;

DECLARE @traceid int = <set your traceid here>;
--stop the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 0;
--add the new event class
EXECUTE sp_trace_setevent @TraceID, 177, 1, 1; -- TextData
EXECUTE sp_trace_setevent @TraceID, 177, 11, 1; -- LoginName
EXECUTE sp_trace_setevent @TraceID, 177, 14, 1; -- StartTime
EXECUTE sp_trace_setevent @TraceID, 177, 21, 1; -- EventSubClass
EXECUTE sp_trace_setevent @TraceID, 177, 23, 1; -- Success
```

```
EXECUTE sp_trace_setevent @TraceID, 177, 34, 1; -- ObjectName
EXECUTE sp_trace_setevent @TraceID, 177, 35, 1; -- DatabaseName
EXECUTE sp_trace_setevent @TraceID, 177, 40, 1; -- DBUserName
EXECUTE sp_trace_setevent @TraceID, 177, 64, 1; -- SessionLoginName
--start the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 1;
```

Audit:

Verify the events captured by the trace:

Run the T-SQL query below:

```
SELECT
    CAST(e.name as varchar(70)) AS Event_Descr,
    CAST(c.name as varchar(70)) AS Column_Descr
FROM sys.fn_trace_geteventinfo(1) t
    INNER JOIN sys.trace_events e
        ON t.eventID = e.trace_event_id
    INNER JOIN sys.trace_columns c
        ON t.columnid = c.trace_column_id
WHERE e.name = 'Audit Server Principal Management Event'
ORDER BY e.name;
```

This query should return the columns listed in the Remediation section.

Default Value:

By default, this information is not captured in the default trace.

References:

<http://www.microsoft.com/sqlserver/en/us/common-criteria.aspx>

1.16.33 Audit Server Scope GDR Event Class (Level 1, Scorable)

Description:

Add the Audit Server Scope GDR Event Class to the audit trace.

Rationale:

This event class indicates that a GDR event has occurred for server permissions.

Remediation:

Stop your trace if running and add the Audit Server Scope GDR Event Class to it:

```
-- Identify your trace by file name with the following query:
SELECT * FROM sys.fn_trace_getinfo(0) WHERE property = 2;

DECLARE @traceid int = <set your traceid here>;
--stop the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 0;
--add the new event class
EXECUTE sp_trace_setevent @TraceID, 170, 1, 1; -- TextData
EXECUTE sp_trace_setevent @TraceID, 170, 11, 1; -- LoginName
EXECUTE sp_trace_setevent @TraceID, 170, 14, 1; -- StartTime
EXECUTE sp_trace_setevent @TraceID, 170, 21, 1; -- EventSubClass
```

```
EXECUTE sp_trace_setevent @TraceID, 170, 23, 1; -- Success
EXECUTE sp_trace_setevent @TraceID, 170, 34, 1; -- ObjectName
EXECUTE sp_trace_setevent @TraceID, 170, 35, 1; -- DatabaseName
EXECUTE sp_trace_setevent @TraceID, 170, 40, 1; -- DBUserName
EXECUTE sp_trace_setevent @TraceID, 170, 64, 1; -- SessionLoginName
--start the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 1;
```

Audit:

Verify the events captured by the trace:

Run the T-SQL query below:

```
SELECT
    CAST(e.name as varchar(70)) AS Event_Descr,
    CAST(c.name as varchar(70)) AS Column_Descr
FROM sys.fn_trace_geteventinfo(1) t
    INNER JOIN sys.trace_events e
        ON t.eventID = e.trace_event_id
    INNER JOIN sys.trace_columns c
        ON t.columnid = c.trace_column_id
WHERE e.name = 'Audit Server Scope GDR Event'
ORDER BY e.name;
```

This query should return the columns listed in the Remediation section.

Default Value:

By default, this information is not captured in the default trace.

References:

<http://www.microsoft.com/sqlserver/en/us/common-criteria.aspx>

1.16.34 Audit Server Starts and Stops Event Class (Level 1, Scorable)

Description:

Add the Audit Server Starts and Stops Event Class to the audit trace.

Rationale:

This event class indicates that the SQL Server service state has been modified.

Remediation:

Stop your trace if running and add the Audit Server Starts and Stops Event Class to it:

```
-- Identify your trace by file name with the following query:
SELECT * FROM sys.fn_trace_getinfo(0) WHERE property = 2;

DECLARE @traceid int = <set your traceid here>;
--stop the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 0;
--add the new event class
EXECUTE sp_trace_setevent @TraceID, 18, 11, 1; -- LoginName
EXECUTE sp_trace_setevent @TraceID, 18, 14, 1; -- StartTime
EXECUTE sp_trace_setevent @TraceID, 18, 21, 1; -- EventSubClass
```



```
EXECUTE sp_trace_setevent @TraceID, 18, 23, 1; -- Success
EXECUTE sp_trace_setevent @TraceID, 18, 64, 1; -- SessionLoginName
--start the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 1;
```

Audit:

Verify the events captured by the trace:

Run the T-SQL query below:

```
SELECT
    CAST(e.name as varchar(70)) AS Event_Descr,
    CAST(c.name as varchar(70)) AS Column_Descr
FROM sys.fn_trace_geteventinfo(1) t
    INNER JOIN sys.trace_events e
        ON t.eventID = e.trace_event_id
    INNER JOIN sys.trace_columns c
        ON t.columnid = c.trace_column_id
WHERE e.name = 'Audit Server Starts and Stops Event'
ORDER BY e.name;
```

This query should return the columns listed in the Remediation section.

Default Value:

By default, this information is captured in the default trace.

References:

<http://www.microsoft.com/sqlserver/en/us/common-criteria.aspx>

1.16.35 Audit Database Operation Event Class (Level 1, Scorable)

Description:

Add the Audit Database Operation Event Class to the audit trace.

Rationale:

This event class indicates that various operations such as checkpoint or subscribe query notification have occurred.

Remediation:

Stop your trace if running and add the Audit Database Operation Event Class to it:

```
-- Identify your trace by file name with the following query:
SELECT * FROM sys.fn_trace_getinfo(0) WHERE property = 2;

DECLARE @traceid int = <set your traceid here>;
--stop the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 0;
--add the new event class
EXECUTE sp_trace_setevent @TraceID, 178, 1, 1; -- TextData
EXECUTE sp_trace_setevent @TraceID, 178, 11, 1; -- LoginName
EXECUTE sp_trace_setevent @TraceID, 178, 14, 1; -- StartTime
EXECUTE sp_trace_setevent @TraceID, 178, 21, 1; -- EventSubClass
```

```
EXECUTE sp_trace_setevent @TraceID, 178, 23, 1; -- Success
EXECUTE sp_trace_setevent @TraceID, 178, 34, 1; -- ObjectName
EXECUTE sp_trace_setevent @TraceID, 178, 35, 1; -- DatabaseName
EXECUTE sp_trace_setevent @TraceID, 178, 40, 1; -- DBUserName
EXECUTE sp_trace_setevent @TraceID, 178, 64, 1; -- SessionLoginName
--start the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 1;
```

Audit:

Verify the events captured by the trace:

Run the T-SQL query below:

```
SELECT
    CAST(e.name as varchar(70)) AS Event_Descr,
    CAST(c.name as varchar(70)) AS Column_Descr
FROM sys.fn_trace_geteventinfo(1) t
    INNER JOIN sys.trace_events e
        ON t.eventID = e.trace_event_id
    INNER JOIN sys.trace_columns c
        ON t.columnid = c.trace_column_id
WHERE e.name = 'Audit Database Operation Event'
ORDER BY e.name;
```

This query should return the columns listed in the Remediation section.

Default Value:

By default, this information is not captured in the default trace.

References:

<http://www.microsoft.com/sqlserver/en/us/common-criteria.aspx>

1.16.36 Audit Schema Object Management Event Class (Level 1, Scorable)

Description:

Add the Audit Schema Object Management Event Class to the audit trace.

Rationale:

This event class indicates that a server object has been created, altered, or dropped.

Remediation:

Stop your trace if running and add the Audit Schema Object Management Event Class to it:

```
-- Identify your trace by file name with the following query:
SELECT * FROM sys.fn_trace_getinfo(0) WHERE property = 2;

DECLARE @traceid int = <set your traceid here>;
--stop the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 0;
--add the new event class
EXECUTE sp_trace_setevent @TraceID, 131, 1, 1; -- TextData
EXECUTE sp_trace_setevent @TraceID, 131, 11, 1; -- LoginName
```

```
EXECUTE sp_trace_setevent @TraceID, 131, 14, 1; -- StartTime
EXECUTE sp_trace_setevent @TraceID, 131, 21, 1; -- EventSubClass
EXECUTE sp_trace_setevent @TraceID, 131, 23, 1; -- Success
EXECUTE sp_trace_setevent @TraceID, 131, 34, 1; -- ObjectName
EXECUTE sp_trace_setevent @TraceID, 131, 35, 1; -- DatabaseName
EXECUTE sp_trace_setevent @TraceID, 131, 40, 1; -- DBUserName
EXECUTE sp_trace_setevent @TraceID, 131, 59, 1; -- ParentName
EXECUTE sp_trace_setevent @TraceID, 131, 64, 1; -- SessionLoginName
--start the trace
EXECUTE sp_trace_setstatus @traceid = @traceid, @status = 1;
```

Audit:

Verify the events captured by the trace:

Run the T-SQL query below:

```
SELECT
    CAST(e.name as varchar(70)) AS Event_Descr,
    CAST(c.name as varchar(70)) AS Column_Descr
FROM sys.fn_trace_geteventinfo(1) t
    INNER JOIN sys.trace_events e
        ON t.eventID = e.trace_event_id
    INNER JOIN sys.trace_columns c
        ON t.columnid = c.trace_column_id
WHERE e.name = 'Audit Schema Object Management Event'
ORDER BY e.name;
```

This query should return the columns listed in the Remediation section.

Default Value:

By default, this information is not captured in the default trace.

References:

<http://www.microsoft.com/sqlserver/en/us/common-criteria.aspx>

1.17 Backup and Disaster Recovery

1.17.1 Backup Strategy (Level 1, Not Scorable)

Description:

Create and follow a backup strategy, taking regular full database backups of all databases (except tempdb) and additional differential and/or transaction log backups. Note that only transaction log backups provide the ability of point-in-time database restore; other backups allow restoring to the time the backup was taken.

Rationale:

Following a backup strategy ensures that no unacceptable data loss can happen and it provides a baseline for auditing and monitoring.

Remediation:

Create and follow a backup strategy.

Audit:

Verify if there is a backup strategy and it is followed.

Default Value:

Not Applicable

References:

1.17.2 Take Regular Full Backups (Level 1, Scorable)

Description:

Regularly take full backups of the databases.

Rationale:

Regular full backups protect you from data loss.

Remediation:

Set up scheduled backup jobs either in SQL Server or in external backup tools.

Audit:

Run the following T-SQL query to check when the last full backup was taken from the databases:

```
SELECT d.name as DBName,
       ISNULL(CONVERT(varchar(16), MAX(bs.backup_finish_date), 120), '-----
       NEVER') as LastBackup
FROM sys.sysdatabases d
LEFT OUTER JOIN msdb.dbo.backupset bs ON bs.database_name = d.name
WHERE d.name != 'tempdb'
GROUP BY d.Name
ORDER BY d.Name;
```

Default Value:

By default, no backups are configured in SQL Server.

References:

<http://blog.rollback.hu/2011/12/verify-latest-database-backups-t-sql/>

1.17.3 Backing up Master Database (Level 1, Not Scorable)

Description:

Backup the master database when any of the following events occur:

- A database is created or deleted.
- Login accounts are created, deleted, or modified
- Server-wide or database settings are modified.

Rationale:

The `master` db contains server level information and must be backed up on a regular basis to ensure recoverability of an instance.

Remediation:

If you are not currently including the backup of the `master` database in your database maintenance plans, do so immediately.

Audit:

Run the following code snippet to determine the last time a backup was performed on the master database.

```
SELECT MAX(backup_finish_date)
FROM msdb.dbo.backupset
WHERE database_name = 'master';
```

Default Value:

The `master` database is not backed up by default.

References:

1.17.4 Backing up MSDB database (Level 1, Not Scorable)

Description:

Backup the `msdb` database when an alert, job, job schedules or operators have been created, modified, or removed.

Rationale:

The `msdb` database holds information regarding SQL Server agent jobs along with backup and restore information. Regular backups of this database should be performed to ensure recoverability of an instance.

Remediation:

If you are not currently including the backup of the `msdb` database in your database maintenance plans, do so immediately.

Audit:

Run the following code snippet to determine the last time a backup was performed on the `msdb` database.

```
SELECT MAX(backup_finish_date)
FROM msdb.dbo.backupset
WHERE database_name = 'msdb';
```

Default:

The `msdb` database is not backed up by default.

References:

<http://msdn.microsoft.com/en-us/library/ms178028.aspx>

1.17.5 *Restrict access to backup files to System Administrators (Level 1, Scorable)*

Description:

Restrict access to the backup files to System Administrators. Secure the backup directory if backup is taken to the file system.

Rationale:

Backup files contain data that is owned by the company. It is essential that backup files are not able to be copied by unauthorized personnel. Unauthorized personnel must not be allowed to modify, move, or delete these files.

Remediation:

Remove any permissions for users who are not system administrators. Grant access to SQL Server service account and database administrators to the backup files.

Audit:

Review the security for the backup destination folders. Review user groups and members that have access to these folders.

Default:

Not Applicable

References:

1.17.6 *Off-server Backup (Level 1, Not Scorable)*

Description:

Keep database backup files or copies of them on another system.

Rationale:

In case the server is physically damaged or the data on the volumes get corrupted, the only way to restore data is an off-server backup.

Remediation:

If you take backups to local disk, copy files to another server or take tape backup of them.

Audit:

Verify if all database backups are available outside of the server.

Default Value:

Not Applicable

References:

1.17.7 *Off-site Backup (Level 2, Not Scorable)*

Description:

Keep database backup files or copies of them in another location than the server is hosted.

Rationale:

In case of a site-wide incident, e.g. natural disaster, the only way to restore data is to use an off-site copy of backup sites.

Remediation:

Copy backup files to another location, preferably to a greater distance.

Audit:

Verify if all database backups are available outside of the current hosting facility.

Default Value:

Not Applicable

References:

1.17.8 *Encrypt Backups (Level 2, Not Scorable)*

Description:

In case a shared backup system is used, encrypt backups containing sensitive data.

Rationale:

Encrypting backups in a shared environment ensures that unauthorized persons cannot gain access to sensitive data even with access to backups.

Remediation:

Encrypt database backups and other sensitive files either with third party tools or custom methods.

Audit:

Verify if all sensitive backups to be backed up from the server are encrypted.

Default Value:

By default, SQL Server can't create encrypted backups.

References:

1.17.9 *Regular Restore Test (Level 1, Not Scorable)*

Description:

Perform regular restore test of database backups.

Rationale:

The only reliable way to verify the correctness of a database backup is to restore it.

Remediation:

Perform a restore test of database backups after setting up backups and later periodically.

Audit:

Verify that restore tests are performed.

Default Value:

Not Applicable

References:

1.17.10 Regular DBCC CHECKDB (Level 1, Scorable)

Description:

Perform regular consistency check on the databases, preferably before backups.

Rationale:

Consistency checks ensure that the database is consistent both from a physical and logical perspective. It is advisable to take full backups after running a check, to ensure that the backups are taken from a known good state.

Remediation:

Perform DBCC CHECKDB regularly on the databases.

Audit:

Query when DBCC CHECKDB ran last successfully by using T-SQL queries from the reference articles.

Default Value:

Not Applicable

References:

http://sqlserverpedia.com/wiki/Last_clean_DBCC_CHECKDB_date

http://sqlblog.com/blogs/jonathan_kehayias/archive/2009/01/28/when-did-dbcc-checkdb-last-run-on-my-databases.aspx

1.17.11 Encryption Components Backup (Level 1, Scorable)

Description:

Backup components used for encryption functions, such as keys and certificates. Use a strong password for them.

Rationale:

Having a backup of the keys and certificates used for encryption is necessary for performing an out-of-place restore of the encrypted database.

Remediation:

Use the BACKUP MASTER KEY, BACKUP SERVICE MASTER KEY and BACKUP CERTIFICATE commands to back up the components. Include them in the backup strategy. Store the backups off-server.

Audit:

Verify if you have the keys and certificates backed up and stored on another server.

Default Value:

By default, these components are not backed up.

References:

<http://msdn.microsoft.com/en-us/library/ms178578%28v=SQL.100%29.aspx>

<http://msdn.microsoft.com/en-us/library/ms174387%28v=SQL.100%29.aspx>

1.18 Replication

1.18.1 *Replication Enabled (Level 1, Scorable)*

Description:

Do not enable replication unless it is in use. If it is required, follow the recommendations of this section

Rationale:

SQL Server replication requires enhanced security settings between servers that share in the replication topology. If not configured properly it could lead to a potential security risk.

Remediation:

Replication should be disabled if it is currently enabled and not being used. Remove replication components by applying the following T-SQL script:

```
DECLARE @distributionDB AS sysname = 'distribution';
DECLARE @publisher AS sysname = 'PublisherServer';
DECLARE @publicationDB as sysname = 'PublicationDB';

-- repeat the two lines below for all databases participating in replication.
USE PublicationDB;
EXECUTE sp_removedbreplication @publicationDB;

USE master
EXECUTE sp_dropdistpublisher @publisher;
EXECUTE sp_dropdistributiondb @distributionDB;
EXECUTE sp_dropdistributor;
```

Audit:

Run the following code snippet to determine if distribution is setup for transactional replication (required):

```
SELECT name,
```

```
is_published,  
is_merge_published,  
is_distributor  
FROM sys.databases;
```

If transactional replication is enabled, the installed column will contain a value of 1 in the result set.

Default Value:

Replication is not enabled by default.

References:

<http://msdn.microsoft.com/en-us/library/ms147921%28v=SQL.100%29.aspx>

1.18.2 *Replication administration roles (Level 1, Not Scorable)*

Description:

Avoid modifying replication administration permissions assigned to the roles by default. Only assign authorized application administrators and DBAs to these roles. The permissions needed to support and administer replication are assigned to sysadmin, db_owner and replmonitor role in distribution database by default.

Rationale:

Replication administration allows to continuously replicate data to an arbitrary server available from the server. This might lead to information disclosure.

Remediation:

Review the logins allowed administering replication. Ensure that all of them is responsible for administering replication. Remove those not requiring this additional permission set.

Audit:

Review the logins allowed administering replication. Ensure that all of them is responsible for administering replication.

Default Value:

By default, the sa account, db_owner in which the publication resides, and replmonitor are able to administer replication.

References:

<http://msdn.microsoft.com/en-us/library/ms152762%28v=SQL.100%29.aspx>

1.18.3 *Replication Agent Account (Level 2, Scorable)*

Description:

Configure replication agents to use a separate Windows account rather than the SQL Server Agent account. Grant only the required permissions to each agent.

Rationale:

Keeping the replication agents separated provides higher level of protection to the SQL Server Agent service. Using a Windows account provides higher security and better account management.

Remediation:

Ensure that all SQL Server replication agents use a Windows account. For those that do not, setup specific Windows accounts to be used for the agents. For changing the accounts, refer to the articles in the references section below.

Audit:

Replication agents are programs that are called through the use of SQL Agent jobs. The properties of these jobs may be viewed by viewing SQL Agent job properties. To view the properties of a SQL Agent Job perform the following steps:

1. Open SQL Server Management Studio.
2. Open Object Explorer and connect to the target instance, which in this case should be the distributor.
3. Navigate to the SQL Server Agent dropdown.
4. The jobs are listed under the jobs folder. You may right click and view the properties of each job here.

Default Value:

The security of these agents is configured when replication publications and subscriptions are created.

References:

<http://msdn.microsoft.com/en-us/library/ms147827%28v=SQL.100%29.aspx>

<http://msdn.microsoft.com/en-us/library/ms151761%28v=SQL.100%29.aspx>

1.18.4 Secure Replication Snapshot Folder (Level 1, Scorable)

Description:

Make available the snapshot folder, which houses a snapshot of the replicated changes, on an explicit share and not an administrative share. Limit NTFS permissions to SQL Server service accounts and database administrators.

Remediation:

Storing snapshot folders on an explicit share allows you to easily assign explicit permissions to the replication agent to access and doesn't require granting administrator permissions on the server. Limiting NTFS permissions prevents unauthorized access to plain text database dump.

Audit:

Review permissions on the network share and NTFS folder where snapshot files are created.

Default Value:

Not applicable.

References:

<http://msdn.microsoft.com/en-us/library/ms151151%28v=SQL.100%29.aspx>

1.18.5 *Secure Replication Over the Internet (Level 1, Not Scorable)*

Description:

Use secure connections, such as VPN, for all replication over the Internet or any shared network. For merge replication, consider using the Web Synchronization.

Rationale:

Using a secure communication connection will ensure that replicated data is not compromised or captured while it is en route.

Remediation:

Consider switching to a secure communication channel, such as VPN, if you are transmitted replicated data over the internet.

Audit:

Review network security to determine if replication is using a public network.

Default Value:

By default, replication does not require a secure communication channel to transmit data.

References:

[http://msdn.microsoft.com/en-us/library/ms151172\(v=SQL.100\).aspx](http://msdn.microsoft.com/en-us/library/ms151172(v=SQL.100).aspx)

1.18.6 *Filtering Replication (Level 1, Not Scorable)*

Description:

Employ replication filters to protect the data where applicable. Use replication filters to bypass replicating sensitive data, such as passwords, social security numbers, payment information, etc. or details of sensitive customers, contracts, etc. SQL Server allows you using both horizontal and vertical filtering.

Rationale:

Filtering out sensitive data minimizes the surface area for that information.

Remediation:

Evaluate the data that you are currently replicating or intend to replicate. Consider using filters so that sensitive columns or rows are not replicated.

Audit:

Review your current replication topology to identify those tables and columns that contain sensitive information. Ensure that those columns are not replicated.

Default Value:

By default, when a table is replicated all fields are transferred. Filters act like a view in that you can specify which fields you want to replicate.

References:

[http://msdn.microsoft.com/en-us/library/ms151775\(v=SQL.100\).aspx](http://msdn.microsoft.com/en-us/library/ms151775(v=SQL.100).aspx)

1.18.7 *Securing Distribution Databases (Level 1, Not Scorable)*

Description:

Distribution databases must be protected equally to the publishing databases.

Rationale:

The distribution database holds a momentary copy of data to be distributed to subscribers. Precautions must be taken to ensure this data is not compromised.

Remediation:

Validate the security for the distribution instance. Take any necessary measures to safeguard against attack.

Audit:

Review the SQL Server security present in the distribution database.

Default Value:

Not applicable.

References:

[http://msdn.microsoft.com/en-us/library/ms151227\(v=SQL.100\).aspx](http://msdn.microsoft.com/en-us/library/ms151227(v=SQL.100).aspx)

1.18.8 *Securing Subscription Databases (Level 1, Not Scorable)*

Description:

Subscription databases must be protected equally to the publishing databases.

Rationale:

The subscription database holds a copy of distributed data. Precautions must be taken to ensure this data is not compromised.

Remediation:

Validate the security for the subscriber instance. Take any necessary measures to safeguard against attack.

Audit:

Review the SQL Server security present in the subscription database.

Default Value:

Not applicable.

References:

1.19 Application Development

1.19.1 *Use Recent Driver (Level 1, Not Scorable)*

Description:

Use a recent database driver supporting important features of SQL Server such as connection encryption and database mirroring.

Rationale:

Using a recent driver allows using all the features of SQL Server 2008.

Remediation:

Ensure that the application is using a recent database driver supporting the features of SQL 2008.

Audit:

Ensure that the application is using a recent database driver supporting the features of SQL 2008.

Default Value:

Not applicable.

References:

1.19.2 *Error Handling (Level 1, Not Scorable)*

Description:

Do not propagate errors back to the user. Log database related errors and/or transmit them to the system administrator.

Rationale:

Displaying system error messages to the user can give details of the underlying database system.

Remediation:

Ensure that the application displays only friendly error messages and that the application logic never bubbles error messages to the user. Instead, the errors should be logged to a table in the database or on the application server and a friendly error message should be displayed.

Audit:

Check with the application teams to ensure only friendly error messages are displayed to the user.

Default Value:

By default, most application layers will display the actual database error message.

References:

1.19.3 *Sanitize User Input (Level 1, Not Scorable)*

Description:

Prevent SQL injection by validating all user input on the application level before transmitting it to the database server.

Rationale:

Using stored procedures parameterizes user input, drastically minimizing any risk for SQL injection.

Remediation:

Consider using stored procedures exclusively for all database interaction. 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.

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.

Default Value:

Not applicable.

References:

<http://msdn.microsoft.com/en-us/library/ms161953%28v=SQL.100%29.aspx>

1.19.4 *Customer Awareness to SQL Express (Level 1, Not Scorable)*

Description:

Inform your customers explicitly that your product includes SQL Server Express Edition so that they can be prepared to install or accept SQL Express specific software updates.

Rationale:

Customers should be made aware of any SQL Express install so that the necessary measures may be taken to ensure data security.

Remediation:

Ensure product documentation includes SQL Express verbiage. In addition, include some safeguarding steps to secure SQL Express data in the documentation.

Audit:

Take an audit of all applications that include SQL Server Express Edition. Ensure that product documentation includes copy that indicates that software operates on SQL Server Express.

Default Value:

Not applicable.

References:

1.19.5 CLR UNSAFE_ACCESS Permission Set (Level 1, Scorable)

Description:

In case you need the clr integration for any application, do not use assemblies with UNSAFE permission set. Consider using assemblies requiring EXTERNAL_ACCESS or UNSAFE_ACCESS permission set outside of the scope of the database engine.

Rationale:

Assemblies with UNSAFE permission set can access external resources such as local and network files or registry as the SQL Server service account. In addition they can read and modify data in the SQL Server process space itself. This poses a threat both from a security and stability point of view.

Remediation:

Use the CREATE ASSEMBLY statement with the WITH PERMISSION_SET = SAFE or EXTERNAL_ACCESS clause.

Audit:

Run the following T-SQL query:

```
SELECT name,  
       permission_set_desc  
FROM sys.assemblies  
where is_user_defined = 1;
```

All the returned assemblies should show SAFE_ACCESS or EXTERNAL_ACCESS in the permission_set_desc column.

Default Value:

By default, the CREATE ASSEMBLY statement assumes the SAFE permission set.

References:

1.19.6 CLR SAFE_ACCESS Permission Set (Level 2, Scorable)

Description:

In case you need the clr enabled option set to true, use only assemblies with SAFE_ACCESS permission set. Consider using assemblies requiring EXTERNAL_ACCESS or UNSAFE_ACCESS permission set outside of the scope of the database engine.

Rationale:

Assemblies with more permission that SAFE_ACCESS can access external resources such as local and network files or registry as the SQL Server service account. This poses a security risk.

Remediation:

Use the CREATE ASSEMBLY statement only with the WITH PERMISSION_SET = SAFE clause.

Audit:

Run the following T-SQL query:

```
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.

Default Value:

By default, the CREATE ASSEMBLY statement assumes the SAFE permission set.

References:

1.19.7 Plain Text Password (Level 1, Scorable)

Description:

Do not store plain text password in application configuration files.

Rationale:

Plain text password in configuration files might be accessible by application users and might give them a wider access to data than necessary.

Remediation:

Consult application administrators to know if there's a password used for connection and if yes, how is it stored. Change the application if it's stored in plain text.

Audit:

Consult application administrators to know if there's a password used for connection and if yes, how is it stored.

Default Value:

By default, no monitoring is set up for the operating system.

References:

1.20 TBD: DROPPED SECTIONS/RECOMMENDATIONS

This section contains the sections/recommendations I deliberately dropped compared to the SQL 2005 one. This section is here only for the consensus phase and should be removed from the final version of the benchmark.

1.20.1 *Permission to Start SQL Writer Service (Level 1, Scorable)*

Description:

Allows backup and restore applications to operate in the Volume Shadow Copy Service (VSS) framework.

Rationale:

This service is required when 3rd party applications are used to make backups of SQL Server data files. Without the use of this service, these applications would fail to take backups of the data files because the SQL Server database engine exclusively locks these files while the service is running.

Remediation:

If you use a 3rd party backup application to take backups of your SQL Server database files, this service is required. Otherwise this service may be disabled. Perform the steps in the Audit section to determine if this service is currently enabled.

Audit:

Perform the following steps to view the details of this service.

1. Run `services.msc`
2. Locate the SQL Server VSS Writer service.
3. Right-click this service to view the properties.
4. If this service is required, set the Startup Type to "Automatic". If this service is stopped, click the "Start" button to enable it.
5. If this service is not required, ensure that the Startup Type is set to "Disabled".

Default Value:

This service is enabled by default and starts automatically when the server starts.

References:

<http://msdn.microsoft.com/en-us/library/ms175536.aspx>

<http://msdn.microsoft.com/en-us/library/ms143504.aspx>

1.20.2 *Permission to read the Remote Procedure Call service (Level 1, Scorable)*

Description:

Grant , the SQL Server service account permission to read the Remote Procedure Call service.

Rationale:

In order to function properly, the SQL Server service account needs permission to read the Remote Procedure Call service.

Remediation:

Audit:

Default Value:

References:

1.20.3 *Permission to read the Event Log service (Level 1, Scorable)*

Description:

Grant , the SQL Server service account permission to read the Event Log service.

Rationale:

In order to function properly, the SQL Server service account needs permission to read the Event Log service.

Remediation:

Audit:

Default Value:

References:

1.20.4 *Permission to Start SQL Server Active Directory Helper Service (Level 1, Scorable)*

Description:

Publishes and manages SQL Server services in Active Directory.

Rationale:

This service publishes SQL Server related information to Active Directory. If this is not required, this service should remain disabled.

Remediation:

If it is necessary to find SQL Server instances from Active Directory, ensure that this service is turned on and set to auto-start. Follow the steps below in the Audit section to enable/disable this service.

Audit:

Perform the following steps to view the details of this service.

Comment [BEG14]: FOR CONSENSUS:
Do we need these here? I'd drop these additional permissions as it can't be converted into a restriction.

1. Run `services.msc`
2. Locate the SQL Server Active Directory Helper Service
3. Right-click this service to view the properties.
4. If this service is required, set the Startup Type to “Automatic”. If this service is stopped, click the “Start” button to enable it.
5. If this service is not required, ensure that the Startup Type is set to “Disabled”.

Default Value:

This service is disabled by default.

References:

<http://msdn.microsoft.com/en-us/library/ms143504.aspx>

1.20.5 Web Task Extended Stored procedures

Removed from SQL 2008.

1.20.6 SQL Mail Extended Procedures

Recommendation is to keep it disabled and only `sp_processmail` is especially dangerous – that one is covered separately.

1.20.7 OLE Automation Stored Procedures

No special thing: if you grant any login the appropriate permissions, it can use it. Not sure why the sysadmin hype was around it – probably old SQL 2000 business.

A

Appendix A: References

1. CIS - Security Benchmark for SQL 2005 v1.2.
2. Microsoft: SQL Server 2008 Security Overview for Database Administrators (as seen at <http://download.microsoft.com/download/a/c/d/acd8e043-d69b-4f09-bc9e-4168b65aaa71/SQL2008SecurityOverviewforAdmins.docx> on 28 May 2012)
3. Microsoft SQL Server 2008 Books Online.
4. SQL Server 2008 R2 Security Best Practices – Operational and Administrative Tasks (as seen at [http://download.microsoft.com/download/1/2/A/12ABE102-4427-4335-B989-5DA579A4D29D/SQL Server 2008 R2 Security Best Practice Whitepaper.docx](http://download.microsoft.com/download/1/2/A/12ABE102-4427-4335-B989-5DA579A4D29D/SQL%20Server%202008%20R2%20Security%20Best%20Practice%20Whitepaper.docx) on 28 May 2012)
5. SQL Server 2005 Database Security Checklist, Version 8 (as seen at

Comment [b15]: Go here to generate a reference
<http://www.neilstoolbox.com/bibliography-creator/reference-book.htm>

http://iase.disa.mil/stigs/downloads/zip/unclassified_sqlserver9_v8r1.7_checklist_20100827.zip on 28 May 2012)

6.

Appendix B: Change History

Date	Version	Changes for this version
TBD, 2012	1.0	Public Release