the CENTER for INTERNET SECURITY

Security Configuration Benchmark For

Microsoft SQL Server 2005

Version 2.0.0

December 31, 2011

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 it 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

Overview

This document, *Security Configuration Benchmark for Microsoft SQL Server 2005*, provides prescriptive guidance for establishing a secure configuration posture for Microsoft SQL Server 2005 SP3 running on Microsoft Windows Server 2003. This guide was tested against Microsoft SQL Server 2005 SP3 on Microsoft Windows Server 2003. 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 guide 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 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 document 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 2005 on Microsoft Windows Server 2003.

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
Tim Chapman, *SQL Server MVP*

Maintainers
Al Comeau, *Microsoft Corporation*
Eric Bitemo

Contributors and Reviewers
Shailesh Athalye*, Symantec, Inc.*
Eric Bitemo
Sheila Christman, *U.S Navy*
Michael Janicke, *Federal Aviation Administration, ATO*
Jignesh Patel
Kevin Zhang, *Continental Airlines*

The CIS community also extends special thanks to Mike Taber of Moon River Consulting, Inc. for authoring the original version of this benchmark, along with the following contributors to previous versions:

Michael Fowkes
Al Comeau
Phyllis R. Palmer
Dana Hemlock
Rajendra Modak
Paul Davis
Mike Chapple
David W. Blaine
Jitesh Chanchani
Dave Shackleford
Balaji Devarasetty
Tran Thanh Chien
Sheila Christman
Michael A. Davis
Ernesto Rojas
Alexey Stolpovskikh
James Hayes
John Thorpe
Drew Miners
William Edmond Jr.
Tyler Harding
John Banghart
Michael Anderson
Carl Alcindor
Brian Lawton
Andrea J. Weber
Michael Mychalczuk
Jannine Mahone
Blake Frantz
Shailesh Athalye

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

1.1 Operating System and Network Specific Configuration

1.1.1 *SQL Servers accessed via Internet (Level 1, Not Scorable)*

Description:
If the SQL Server is being accessed via the Internet, place the SQL Server inside a DMZ.

Rationale:
Deploying SQL behind a firewall will reduce the remote attack surface of the underlying OS and SQL related services.

Remediation:
If the SQL Server is not inside a DMZ consider doing so. Ensure the firewall blocks inbound requests to TCP port 1433 and UDP port 1434 as well as any other ports associated with named instances.

Audit:
Perform a network audit to determine if your SQL Server resides inside a DMZ with the web server.

Default Value:
No applicable.

References:
http://www.windowsecurity.com/articles/Secure_Architecture_SQL_Web_Server.html

1.1.2 *Ensure Data in Transit is Encrypted (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 SSL be leveraged. It is also recommended to use the fully-qualified DNS name of the server in the SSL certificate to help prevent masquerading.

Rationale:
Enabling SSL between Microsoft SQL Server and its clients will protect the confidentiality and integrity of the information transmitted between them.

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 Computer Manager using the following Run command: `compmgmt.msc`.
2. Expand Services and Applications.
3. Expand SQL Server Configuration Manager.

4. Expand SQL Server Network Configuration and right click Protocols for SQL Server; select Properties.
5. The Force Encryption option may be set on the Flags tab.

Audit:
To perform the following steps to determine if SQL Server is using encrypted connections:

1. Open Computer Manager using the following Run command: `compmgmt.msc`.
2. Expand Services and Applications.
3. Expand SQL Server Configuration Manager.
4. Expand SQL Server Network Configuration and right click Protocols for SQL Server; select Properties.
5. The Force Encryption option may be viewed on the Flags tab.

Default Value:
SQL Server connections are not encrypted by default.

References:
http://msdn.microsoft.com/en-us/library/ms189067.aspx

1.1.3 *Segregate Production and Test Servers (Level 1, Not Scorable)*

Description:
Maintain test and development servers on a separate network segment from the production servers. Do not allow linked server connections from the test and development servers to the production servers.

Rationale:
The security posture of test and development servers may be relaxed compared to production instances. If test or development servers are linked to production systems, the risk to the production server is increased.

Remediation:
Segment test and production networks and remove any existing server links between these systems.

Audit:
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 is not natively able to speak to other instances.

References:
http://msdn.microsoft.com/en-us/library/aa213778(SQL.80).aspx

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.

1.1.5 *Disable Internet Information Services (Level 2, Scorable)*

Description:
Disable the Internet Information Services (IIS) web server on systems running SQL Server..

Rationale:
Disabling IIS will reduce the remote attack surface of SQL server.

Remediation:
Perform the following to configure the system as recommended:

1. Execute `services.msc`.
2. Right click the Internet Information Services service.
3. Left click Properties.
4. Set the Startup type to Disabled.

5. Click Apply.

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

1. Execute `services.msc`.
2. Right click the Internet Information Services service.
3. Left click Properties.
4. Verify that the Startup type is set to Disabled.

Default Value:
This service is not installed by default.

1.1.6 *SQL Server Active Directory Helper (Level 1, Scorable)*

Description:
The SQL Server Active Directory Helper publishes SQL Server services in Active Directory. If SQL server is not domain joined, or the ability to find SQL instances via Active Directory is not required, it is recommended that this service be disabled.

Rationale:
This service publishes SQL Server related information to Active Directory. If this is not required, this service should remain disabled as a defense in depth measure.

Remediation:
Perform the following steps to disable this service.

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 not required, ensure that the Startup Type is set to "Disabled".

Audit:
Perform the following steps to configure the system as required view the details of this service.

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

1.1.7 *Permission to start SQL Writer (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
http://technet.microsoft.com/en-us/library/cc737523%28WS.10%29.aspx

1.2 User Accounts

1.2.1 *SQL Server Service Account (Level 2, Not Scorable)*

Description:
Use a least-privileged Local or Domain account for the SQL Server service. 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.

Rationale:
If the MSSQL Server Service is compromised, the attacker's ability to pivot to other systems is reduced if the compromised service is operating under the context of a least privileged principal.

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.

Default Value:
By default the SQL Server service runs under a local user account. This is set when SQL Server is installed.

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

1.2.2 *SQL Server Agent Service Account (Level 1, Not Scorable)*

Description:
If replication, DTS, or other inter-server communication is required, the SQL Server Agent account must be a domain account. Ensure the domain account used for the SQL Server Agent Service is least-privilege.

Rationale:
If the SQL Server Agent Service is compromised, the attacker's ability to pivot to other systems is reduced if the compromised service is operating under the context of a least privileged principal.

Remediation:

Ensure that the SQL Agent uses a low-privilege domain account. Doing so restricts the access of the SQL Server Agent service to network and system 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.

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

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

1.2.3 *Local Administrators group membership (Level 1, Scorable)*

Description:
Any local or domain account used by SQL server services must not be a member of the local Administrators group.

Rationale:
Ensuring that SQL server related services are not member of the local Administrators group will reduce an attacker's ability to compromise other local assets, accounts, services, or information on the server.

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 required administrative permissions.

References:

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

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

Description:
Make a domain service account a member of only non-privileged 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, such as a Power User or Administrator, consider revising the security scheme to encompass only the necessary permissions.

Audit:
The domain user permissions may be viewed through Active Directory.

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

1.3 SQL Server service account rights

Grant the SQL Server service account(s) the following rights:

1.3.1 *Log on as a service (Level 1, Scorable)*

Description:
This security setting determines which service accounts can register a process as a service.

Rationale:
SQL Server runs as a service, so this privilege is required.

Remediation:
When SQL Server is installed, the account that the service runs under is granted this permission. If for some reason this account wasn't granted this permission, the SQL Server service would not start. If this were to occur, perform the Audit outlined in this section and grant this permission to the account running the SQL Server service.

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. Add/remove the necessary users

Default Value:
By default, no accounts have the privilege to log on as a service.

References:
http://technet.microsoft.com/en-us/library/cc957141.aspx

1.3.2 *Act as part of the operating system (Level 1, Scorable)*

Description:
This user right allows a process to impersonate any user without authentication. The process can therefore gain access to the same local resources as that user. Processes that require this privilege should use the `LocalSystem` account, which already includes this privilege, rather than using a separate user account with this privilege specially assigned. If your organization only uses servers that are members of the Windows Server 2003 family, you do not need to assign this privilege to your users. However, if your organization uses servers running Windows 2000 or Windows NT 4.0, you might need to assign this privilege to use applications that exchange passwords in plaintext. Assigning this user right can be a security risk. Only assign this user right to trusted users.

Rationale:
This permission is necessary for SQL Server to operate.

Remediation:
If the SQL Server service account is not a local administrator on the SQL Server machine, it must be granted this permission. Follow the steps outlined in the Audit section to locate this permission and assign the necessary users.

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. Add/remove the necessary users

Default Value:
By default, only the `LocalSystem` account as the privilege to act as part of the operating system.

References:

http://technet.microsoft.com/en-us/library/cc976442.aspx
http://support.microsoft.com/kb/283811

1.3.3 *Log on as a batch job (Level 1, Scorable)*

Description:
This security setting allows a user to be logged on by means of a batch-queue facility. For example, when a user submits a job by means of the task scheduler, the task scheduler logs that user on as a batch user rather than as an interactive user. In Windows 2000 Server, Windows 2000 Professional, Windows XP Professional and the Windows Server 2003 family, the Task Scheduler automatically grants this right as necessary.

Rationale:
SQL Server Agent proxies define the security context for a job step. SQL Server Agent proxies use credentials to store information about Windows user accounts. The user who is specified in the credential must have the "Log on as a batch job" permission on the computer that is running SQL Server 2005.

Remediation:
If the SQL Server Agent service account or proxy account is not a local administrator on the SQL Server machine, it must be granted this permission. Follow the steps outlined in the Audit section to locate this permission and assign the necessary users.

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. Add/remove the necessary users

Default Value:
By default, only the `LocalSystem` account as the privilege to log on as a batch job.

References:
http://technet.microsoft.com/en-us/library/cc957131.aspx
http://support.microsoft.com/kb/912911

1.3.4 *Replace a process-level token (Level 1, Scorable)*

Description:
This security setting determines which user accounts can call the `CreateProcessAsUser()` application programming interface (API) so that one service can start another. An example of a process that uses this user right is Task Scheduler.

Rationale:

The SQL Server services have needs to start other processes, so this privilege is required by the accounts that run these services. One example is the fact that the SQL Server service must first be running before the SQL Server Agent service may start.

Remediation:
SQL Server and the SQL Server Agent services must possess this permission. Follow the steps outlined in the Audit section to locate this permission and assign the necessary users.

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. Add/remove the necessary users

Default Value:
By default, only the `LocalSystem` account and the Network Service Account are given the privilege to replace a process-level token.

References:
http://technet.microsoft.com/en-us/library/cc784623%28WS.10%29.aspx

1.3.5 *Bypass traverse checking (Level 1, Scorable)*

Description:
This user right determines which users can traverse directory trees even though the user may not have permissions on the traversed directory. This privilege does not allow the user to list the contents of a directory, only to traverse directories. This user right is defined in the Default Domain Controller Group Policy object (GPO) and in the local security policy of workstations and servers.

Rationale:
The accounts that run the SQL Server and SQL Server Agent services must be given this privilege so these services are able to view directory structures on the server.

Remediation:
When SQL Server is installed, the account that the service runs under is granted this permission. If for some reason this account wasn't granted this permission, the SQL Server service would not start. Perform the Audit outlined in this section and grant this permission to the account running the SQL Server service.

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. Add/remove the necessary users

Default Value:
On workstations and servers, the following groups are granted this permission: Administrators, Backup Operators, Power Users, Users, Everyone. On Domain Controllers, the following groups are granted this permission: Administrators, Authenticated Users.

References:
http://technet.microsoft.com/en-us/library/cc739389(WS.10).aspx

1.3.6 *Adjust memory quotas for a process (Level 1, Scorable)*

Description:
This privilege determines who can change the maximum memory that can be consumed by a process. This user right is defined in the Default Domain Controller Group Policy object (GPO) and in the local security policy of workstations and servers.

Rationale:
SQL Server internally manages its own memory consumption. This privilege must be assigned to the account that the SQL Server services runs under so that SQL Server is able to perform this memory management.

Remediation:
If the SQL Server service account is not a local administrator on the SQL Server machine, it must be granted this permission. Follow the steps outlined in the Audit section to locate this permission and assign the necessary users.

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. Add/remove the necessary users

Default Value:
By default, only Administrators have this privilege.

References:
http://technet.microsoft.com/en-us/library/cc736528(WS.10).aspx

1.4 SQL Server Agent service account rights

The following rights are granted to the Server Agent service account by default:

1.4.1 *Log on as a service (Level 1, Scorable)*

Description:
This security setting determines which service accounts can register a process as a service.

Rationale:
SQL Server runs as a service, so this privilege is required.

Remediation:
When SQL Server is installed, the account that the service runs under is granted this permission. If for some reason this account wasn't granted this permission, the SQL Server service would not start. If this were to occur, perform the Audit outlined in this section and grant this permission to the account running the SQL Server service.

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. Add/remove the necessary users

Default Value:
By default, no accounts have the privilege to log on as a service.

References:
http://technet.microsoft.com/en-us/library/cc957141.aspx

1.4.2 *Act as part of the operating system (Level 1, Scorable)*

Description:
This user right allows a process to impersonate any user without authentication. The process can therefore gain access to the same local resources as that user. Processes that require this privilege should use the `LocalSystem` account, which already includes this privilege, rather than using a separate user account with this privilege specially assigned. If your organization only uses servers that are members of the Windows Server 2003 family, you do not need to assign this privilege to your users. However, if your organization uses servers running Windows 2000 or Windows NT 4.0, you might need to assign this privilege to use applications that exchange passwords in plaintext. Assigning this user right can be a security risk. Only assign this user right to trusted users.

Rationale:
This permission is necessary for SQL Server to operate if the account that the SQL Server services runs under is not a local administrator on the SQL Server machine.

Remediation:

If the SQL Server service account is not a local administrator on the SQL Server machine, it must be granted this permission. Follow the steps outlined in the Audit section to locate this permission and assign the necessary users.

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. Add/remove the necessary users

Default Value:
By default, only the `LocalSystem` account as the privilege to act as part of the operating system.

References:
http://technet.microsoft.com/en-us/library/cc976442.aspx
http://support.microsoft.com/kb/283811

1.4.3 *Log on as a batch job (Level 1, Scorable)*

Description:
This security setting allows a user to be logged on by means of a batch-queue facility. For example, when a user submits a job by means of the task scheduler, the task scheduler logs that user on as a batch user rather than as an interactive user. In Windows 2000 Server, Windows 2000 Professional, Windows XP Professional and the Windows Server 2003 family, the Task Scheduler automatically grants this right as necessary.

Rationale:
SQL Server Agent proxies define the security context for a job step. SQL Server Agent proxies use credentials to store information about Windows user accounts. The user who is specified in the credential must have the "Log on as a batch job" permission on the computer that is running SQL Server 2005.

Remediation:
If the SQL Server Agent service account or proxy account is not a local administrator on the SQL Server machine, it must be granted this permission. Follow the steps outlined in the Audit section to locate this permission and assign the necessary users.

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. Add/remove the necessary users

Default Value:
By default, only the `LocalSystem` account as the privilege to log on as a batch job.

References:
http://technet.microsoft.com/en-us/library/cc957131.aspx
http://support.microsoft.com/kb/912911

1.4.4 *Replace a process-level token (Level 1, Scorable)*

Description:
This security setting determines which user accounts can call the `CreateProcessAsUser()` application programming interface (API) so that one service can start another. An example of a process that uses this user right is Task Scheduler.

Rationale:
The SQL Server services have needs to start other processes, so this privilege is required by the accounts that run these services. One example is the fact that the SQL Server service must first be running before the SQL Server Agent service may start.

Remediation:
If the SQL Server or SQL Server Agent service account is not a local administrator on the SQL Server machine, it must be granted this permission. Follow the steps outlined in the Audit section to locate this permission and assign the necessary users.

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. Add/remove the necessary users

Default Value:
By default, only the `LocalSystem` account and the Network Service Account are given the privilege to replace a process-level token.

References:
http://technet.microsoft.com/en-us/library/cc784623%28WS.10%29.aspx

1.4.5 *Bypass traverse checking (Level 1, Scorable)*

Description:
This user right determines which users can traverse directory trees even though the user may not have permissions on the traversed directory. This privilege does not allow the user to list the contents of a directory, only to traverse directories. This user right is defined in

the Default Domain Controller Group Policy object (GPO) and in the local security policy of workstations and servers.

Rationale:
The accounts that run the SQL Server and SQL Server Agent services must be given this privilege so these services are able to view directory structures on the server.

Remediation:
When SQL Server is installed, the account that the service runs under is granted this permission. If for some reason this account wasn't granted this permission, the SQL Server service would not start. Perform the Audit outlined in this section and grant this permission to the account running the SQL Server service.

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. Add/remove the necessary users

Default Value:
On workstations and servers, the following groups are granted this permission: Administrators, Backup Operators, Power Users, Users, Everyone. On Domain Controllers, the following groups are granted this permission: Administrators, Authenticated Users.

References:
http://technet.microsoft.com/en-us/library/cc739389(WS.10).aspx

1.4.6 *Adjust memory quotas for a process (Level 1, Scorable)*

Description:
This privilege determines who can change the maximum memory that can be consumed by a process. This user right is defined in the Default Domain Controller Group Policy object (GPO) and in the local security policy of workstations and servers.

Rationale:
SQL Server internally manages its own memory consumption. This privilege must be assigned to the account that the SQL Server services runs under so that SQL Server is able to perform this memory management.

Remediation:
If the SQL Server service account is not a local administrator on the SQL Server machine, it must be granted this permission. Follow the steps outlined in the Audit section to locate this permission and assign the necessary users.

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. Add/remove the necessary users

Default Value:
By default, only Administrators have this privilege.

References:
http://technet.microsoft.com/en-us/library/cc736528(WS.10).aspx

1.5 Integration Service account rights

Grant the Integration Service account(s) the following rights:

1.5.1 *Log on as a service (Level 1, Scorable)*

Description:
This security setting determines which service accounts can register a process as a service.

Rationale:
SQL Server runs as a service, so this privilege is required.

Remediation:
When SQL Server is installed, the account that the service runs under is granted this permission. If for some reason this account wasn't granted this permission, the SQL Server service would not start. If this were to occur, perform the Audit outlined in this section and grant this permission to the account running the SQL Server service.

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. Add/remove the necessary users

Default Value:
By default, no accounts have the privilege to log on as a service.

References:
http://technet.microsoft.com/en-us/library/cc957141.aspx

1.5.2 *Bypass traverse checking (Level 1, Scorable)*

Description:
This user right determines which users can traverse directory trees even though the user may not have permissions on the traversed directory. This privilege does not allow the user to list the contents of a directory, only to traverse directories. This user right is defined in the Default Domain Controller Group Policy object (GPO) and in the local security policy of workstations and servers.

Rationale:
The accounts that run the SQL Server and SQL Server Agent services must be given this privilege so these services are able to view directory structures on the server.

Remediation:
When SQL Server is installed, the account that the service runs under is granted this permission. If for some reason this account wasn't granted this permission, the SQL Server service would not start. Perform the Audit outlined in this section and grant this permission to the account running the SQL Server service.

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. Add/remove the necessary users

Default Value:
On workstations and servers, the following groups are granted this permission: Administrators, Backup Operators, Power Users, Users, Everyone. On Domain Controllers, the following groups are granted this permission: Administrators, Authenticated Users.

References:
http://technet.microsoft.com/en-us/library/cc739389(WS.10).aspx

1.5.3 *Create global objects (Level 1, Scorable)*

Description:
This user right is required for a user account to create global objects during Terminal Services sessions. Users can still create session-specific objects without being assigned this user right.

Rationale:
Assigning this permission can be a security risk. Only assign this permission when absolutely necessary. Users are still able to create session-specific objects without being assigned this user right.

Remediation:

Perform the Audit outlined in this section and grant this permission to the account running the SQL Server service. Remove this permission if it has been unnecessarily granted.

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. Add/remove the necessary users

Default Value:
This permission is granted by default.

References:
http://technet.microsoft.com/en-us/library/cc739176(WS.10).aspx

1.5.4 *Impersonate a client after authentication (Level 1, Scorable)*

Description:
Assigning this privilege to a user allows programs running on behalf of that user to impersonate a client. Requiring this user right for this kind of impersonation prevents an unauthorized user from convincing a client to connect (for example, by remote procedure call (RPC) or named pipes) to a service that they have created and then impersonating that client, which can elevate the unauthorized user's permissions to administrative or system levels.

Rationale:
This permission is typically not needed as the necessary services are given this user right when they are started.

Remediation:
When SQL Server is installed, the account that the service runs under is granted this permission. Perform the Audit outlined in this section and grant this permission to the account running the SQL Server service.

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. Add/remove the necessary users

Default Value:
By default, only administrators are given this right.

References:
http://technet.microsoft.com/en-us/library/cc787897(WS.10).aspx

1.5.5 *SQL Server Agent Proxy accounts (Level 1, Not Scorable)*

Description:
Create dedicated user accounts specifically for proxies, and only use these proxy user accounts for running job steps.

Rationale:
A SQL Server Agent proxy defines the security context for a job step. A proxy provides SQL Server Agent with access to the security credentials for a Microsoft Windows user. Each proxy can be associated with one or more subsystems. A job step that uses the proxy can access the specified subsystems by using the security context of the windows user. Before SQL Server agent runs a job step that uses a proxy, SQL Server Agent impersonates the credential defined in the proxy and then runs the job step by using that security context.

Remediation:
If the SQL Agent service is currently running an administrative domain user, consider changing this user and start using dedicated proxy accounts to ensure that only the necessary domain privileges are granted to this account.

Audit:
Review the domain privileges of the SQL Server agent service. Make revisions for implementing SQL Agent proxy accounts as necessary. These permissions may be viewed in Active Directory.

Default Value:
The proxy account is given no administrative permissions by default.

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

1.6 SQL Server Installation and Patches

1.6.1 *SQL Server install platform (Level 1, 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

1.6.2 *Patches and hot fixes (Level 1, Scorable)*

Description:
SQL Server patches and hot fixes contain program updates that fix found issues in the software.

Rationale:
Patches and hot fixes typically contain specific fixes for security vulnerabilities.

Remediation:
Ensure the Current SQL Server service pack and hot fixes 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 service SQL Server service packs.

Audit:
To determine your SQL Server service pack level, run the following code snippet.

```
SELECT @@VERSION
```

Default Value:
Service packs and patches are not installed by default.

References:
http://support.microsoft.com/sp

1.6.3 *SQL Server Ports (Level 1, Scorable)*

Description:
Change SQL Server default ports from 1433 and 1434.

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 and port 1434 for UDP traffic.

Remediation:
If your SQL Server instances are listening on ports 1433 and 1434, change them to a higher port that is not in use. Doing so can mask SQL Server from various vulnerability tools.

Audit:
SQL Server port assignments may be viewed in the SQL Server configuration manager in the protocols menu.

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

1. Run `compmgmt.msc` from the Run menu.
2. Expand Services and Applications.
3. Expand SQL Server Configuration Manager.
4. Expand SQL Native Client Configuration. Select Client Protocols.
5. Right-click TCP/IP and select Properties. The default value may be set in the Protocol tab.

Default Value:
By default, SQL Server listens on to TCP/IP traffic on port 1433 and to UPD traffic on port 1434.

References:
http://msdn.microsoft.com/en-us/library/ms190944.aspx

1.6.4 *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 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:
Consider renaming your SQL Server instances if the name indicates the purposes of the machine.

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

Default Value:
SQL Server default instances are the same as the machine name.

References:
http://msdn.microsoft.com/en-us/library/aa176583(SQL.80).aspx

1.6.5 *SQL Server instances (Level 1, Not Scorable)*

Description:
Keep an inventory of all versions, editions and languages of SQL Server.

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 meticulously kept. There are some good third party applications that you can use to inventory your SQL Server environment. Such tools include SQL Scan, SQL Ping, and SQL Check.

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 network instances.

References:
http://www.sqlsecurity.com/Tools/FreeTools/tabid/65/Default.aspx

1.6.6 *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:
http://msdn.microsoft.com/en-us/library/aa905171(SQL.80).aspx

1.6.7 *Rename `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.

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 = Admin_User;
```

Audit:
Use the following code snippet to determine if the `sa` account is present and if it is disabled.

```
SELECT name, is_disabled
FROM sys.server_principals
WHERE name = sa;
```

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

1.6.8 *Strong `sysadmin` password (Level 1, Not Scorable)*

Description:
Use a strong password for the `sa` login account.

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.

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:
If you leave the `sa` password blank, SQL Server Management Studio will give you a warning indicating that doing so poses security threats.

References:
http://msdn.microsoft.com/en-us/library/ms345149%28SQL.90%29.aspx

1.6.9 *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:
http://www.microsoft.com/downloads/details.aspx?FamilyID=e719ecf7-9f46-4312-af89-6ad8702e4e6e&displaylang=en

1.6.10 *Initialization parameter (Level 1, Scorable)*

Description:
Enable C2-Audit mode if constant security-level events must be captured that are not captured by the SQL Server default trace.

Rationale:

C2-Audit mode may be used to capture more detailed security data which enables you to more easily diagnose security related violations.

Remediation:
Review the system requirements for capturing audit data. C2 Audit mode captures a large amount of system security activity to allow you to track down security policy violations. The activity of C2 audit mode may also introduces potential performance issues. This option should be enabled only when the benefit of the feature outweigh the performance drawbacks.

Audit:
Use the following code snippet to determine the current setting for C2-Audit mode.

```
EXECUTE sp_configure 'show advanced options',1
RECONFIGURE WITH OVERRIDE
EXECUTE sp_configure 'c2 audit mode'
```

Default Value:
C2-Audit mode is disabled by default.

References:
http://msdn.microsoft.com/en-us/library/ms187634.aspx

1.6.11 *Initialization parameter – Remote Access (Level 1, Scorable)*

Description:
Do not enable Remote Access unless replication is being used or the requirement is justified.

Use the remote access option to control the execution of stored procedures from local or remote servers on which instances of Microsoft SQL Server are running. Leaving this option off prevents local stored procedures from being run from a remote server or remote stored procedures from being run on the local server.

Rationale:
This option is only necessary in those situations where local procedures need to be ran on a remote server or the local server needs to run procedures on remote servers. This setting should not be enabled unless cross server execution is required.

Remediation:
Disable this feature if it is not required.

Audit:
Run the following code snippet to determine if remote access is enabled for the SQL Server instance:

```
EXECUTE sp_configure 'remote access'
```

A value of 1 in the run_value column indicates that the option is enabled. Disable the option by running the following statement:

```
EXECUTE sp configure 'remote access', 1
RECONFIGURE
```

Default Value:
This option is enabled by default.

References:
http://msdn.microsoft.com/en-us/library/ms187660.aspx

1.6.12 *Initialization parameter – Scan for Startup Procedures (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 executing 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 audit traces and other commonly used monitoring SPs from re-starting on start up.

Remediation:
Ensure that this feature is disabled if there are no user-defined stored procedures set to execute when SQL Server starts.

Audit:
Run the following code snippet to determine if SQL Server will scan and execute defined stored procedures at startup:

```
EXECUTE sp_configure 'show advanced options', 1
RECONFIGURE WITH OVERRIDE
EXECUTE sp_configure 'scan for startup procs'
```

To disable this option if it is enabled:

```
EXECUTE sp_configure 'scan for startup procs', 0
RECONFIGURE WITH OVERRIDE
```

Default:
This option is disabled by default.

References:
http://msdn.microsoft.com/en-us/library/ms179460.aspx

1.6.13 *Separate partitions (Level 1, Scorable)*

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

Rationale:
Separate partitions provide greater protections via host and file permissions at the volume level as well as allowing greater control over data storage usage and monitoring of the database.

Remediation:
You should considering moving the data files, log files, and tempdb files to separate partitions if they currently share a partition.

Audit:

Use the following script to determine where all database files are located for a SQL Server instance:

```
SELECT *
FROM sys.sysaltfiles;
```

Default Value:
By default, SQL Server data and log files are placed in the SQL Server Program Files directory where SQL Server installed. This default location may be altered when SQL Server is installed.

References:
http://msdn.microsoft.com/en-us/library/ms179316.aspx

1.7 SQL Server Settings

1.7.1 *SQL Server Configuration Manager (Level 1, Scorable)*

Description:
Ensure the Named Pipes protocol is disabled where not needed. A named pipe is a specifically named one-way or two-way channel for communication between a pipe server and one or more pipe clients. SQL Server checks the named pipe to verify what connections can log on to the system that is running SQL Server to run queries against data that is stored on the server.

Rationale:

In the past, Named Pipes have been susceptible to hijacking which allows the attacker to gain control of another client session. If Named Pipes is required, change the name to something other than `\\.\pipe\sql\query`. Named Pipes protocol is disabled by default for MSSQLSERVER and SQLEXPRESS and enabled for SQL Native Client.

Remediation:
In the event that Named Pipes is enabled and not required, ensure it is disabled. TCP/IP is a good alternative, especially if your application does not require SQL Server authentication.

Audit:
SQL Server port assignments may be viewed in the SQL Server configuration manager in the protocols menu.

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

1. Run `compmgmt.msc` from the Run menu.
2. Expand Services and Applications.
3. Expand SQL Server Configuration Manager.
4. Expand SQL Native Client Configuration. Select Client Protocols.
5. Right click Named Pipes and select Enable where required.

Default Value:
The Named Pipes protocol is disabled by default on a new SQL Server installation.

References:
http://msdn.microsoft.com/en-us/library/aa266530%28VS.60%29.aspx

1.7.2 *Auto Restart SQL Server (Level 1, Scorable)*

Description:
Set the SQL Server service start mode to 'Automatic'

Rationale:
This is found in the SQL Server Configuration Manager. Note: This recommendation is not applicable to clustered SQL instances.

Remediation:
Disable this service if it is currently enabled in a clustered environment. Otherwise, ensure that the service Startup Type is set to Automatic.

Audit:
Perform the following steps to determine the Startup Type of this service:

1. Run `services.msc` from the Run menu.
2. Locate the SQL Server service from the Services menu. The Startup Type may be viewed in the Startup Type column in this menu.

3. To alter the Startup Type, right click the service and select Properties.
4. The Startup Type may be altered from the Startup Type dropdown list.

Default Value:
By default, this service is set to Manual.

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

1.7.3 *Auto Restart SQL Server Agent (Level 1, Scorable)*

Description:
If the SQL Server Agent is required, set the 'SQL Server Agent' start mode to 'Automatic'.

Rationale:
This is found in the SQL Server Configuration Manager. Note: This recommendation is not applicable to clustered SQL instances.

Remediation:
Disable this service if it is currently enabled in a clustered environment. Otherwise, ensure that the service Startup Type is set to Automatic.

Audit:
Perform the following steps to determine the Startup Type of this service:

1. Run `services.msc` from the Run menu.
2. Locate the SQL Server Agent service from the Services menu. The Startup Type may be viewed in the Startup Type column in this menu.
3. To alter the Startup Type, right click the service and select Properties.
4. The Startup Type may be altered from the Startup Type dropdown list.

Default Value:
By default, this service is set to Manual.

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

1.7.4 *Distributed Transaction Coordinator (Level 1, Scorable)*

Description:
Set the 'Distributed Transaction Coordinator' service start mode to 'Disabled' if this service is not required.

Rationale:
This is found in the SQL Server Configuration Manager.

Remediation:
Ensure that the service Startup Type is set to Disabled if the DTC is not required..

Audit:
Perform the following steps to determine the Startup Type of this service:

1. Run `services.msc` from the Run menu.
2. Locate the Distributed Transaction Coordinator service from the Services menu. The Startup Type may be viewed in the Startup Type column in this menu.
3. To alter the Startup Type, right click the service and select Properties.
4. The Startup Type may be altered from the Startup Type dropdown list.

Default Value:
By default, this service is set to Manual.

References:
http://ss64.com/nt/syntax-services.html

1.7.5 *Cross database-ownership chaining (Level 1, Scorable)*

Description:
Ownership chaining enables managing access to multiple objects, such as multiple tables, by setting permissions on one object, such as a view. Ownership chaining also offers a slight performance advantage in scenarios that allow for skipping permission checks.

Rationale:
Database ownership chaining may allow for non-explicit and unchecked access for common owners of objects between databases. Keeping this feature disabled mitigates this potential threat.

Use `sp_dboption` to check for databases for which cross-database ownership chaining is enabled. This is found in the General page of SQL Server Properties window.

Remediation:
Disable cross database ownership chaining if it is enabled and not required.

Audit:
Run the following script to determine if database ownership chaining is on for a given database:

```
SELECT name, is_db_chaining_on
FROM sys.databases
WHERE name = '<your database name>';
```

Default Value:
Database ownership chaining is disabled by default and must explicitly be setup between specific databases to be used.

References:
http://msdn.microsoft.com/en-us/library/ms188676.aspx

1.7.6 *Advanced Server Settings (Level 1, Scorable)*

Description:
Do not allow direct updates to the SQL Server system catalog tables.

Rationale:
Direct system catalog modifications are not available in SQL Server 2005. This access level is disabled by default in SQL Server 2005 and cannot be enabled. You must use the documented API's to access them.

Remediation:
No remediation required. System updates are not allowed in SQL Server 2005 and later.

Audit:
Run the following code snippet to view that updates to system catalogs are not allowed.

```
EXECUTE sp_configure 'allow updates';
```

Note that even if this setting is enabled system catalogs will still now allow updates.

Default Value:
System updates are not allowed in SQL Server 2005.

References:
http://msdn.microsoft.com/en-us/library/ms186267(SQL.90).aspx

1.7.7 *Backup/Restore from tape timeout (Level 1, Scorable)*

REMOVE FROM BENCHMARK. THIS POINT DOESN'T MAKE SENSE TO BE INCLUDED.

1.7.8 *Media Retention (Level 1, Not Scorable)*

Description:
Set the default backup media retention to the minimum number of days needed to retain a full backup of the database. Ideally, this should be as high as your resources permit.

Rationale:
Keep as many days of full database backups as possible. This allows you to restore a certain day of the database if required.

Remediation:
Additional full backups should be stored if the media is available to do so.

Audit:
Determine the available amount of space that is available that may be devoted to storing backup files.

Default:
Database backups are not removed by default. Retention of backups must be setup in the maintenance plan.

1.7.9 *Data Directory (Level 1, Scorable)*

Description:
The default data directory should be a dedicated data partition.

Rationale:
The data files should be separated on a dedicated partition for performance and security purposes. This data partition would optimally be a RAID 10 partition.

Remediation:
If the data files share a drive with the OS and/or the SQL log files they should be moved to their own partition. View the data properties for data file placement.

Audit:
Use the following code snippet to determine file locations for all databases on the SQL Server instance:

```
SELECT
      DatabaseName =  name,
      FileLocation = filename
FROM sys.sysaltfiles
ORDER BY name;
```

Default Value:
All SQL Server files are placed in the [drive]\Program Files\Microsoft SQL Server installation directory. This can be altered at installation and the file placement may be specified when new databases are created.

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

1.7.10 *Log File Directory (Level 1, Scorable)*

Description:
The default log directory should be a dedicated partition separate from all programs and data.

Rationale:
SQL Server log files are written to frequently and sequentially. System performance will benefit by placing the log files on their own dedicated drive partition.

Remediation:
Review log file placement for all user defined databases. Consider placing log files on their own partition if resources allow.

Audit:
Use the following code snippet to determine file locations for all databases on the SQL Server instance:

```
SELECT
      DatabaseName =  name,
      FileLocation = filename
FROM sys.sysaltfiles
ORDER BY name;
```

Default Value:
All SQL Server files are placed in the `%ProgramFiles%\Microsoft SQL Server` installation directory. This can be altered at installation and the file placement may be specified when new databases are created.

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

1.7.11 *Replication (Level 1, Scorable)*

Description:
Do not enable replication.

Section 7 covers security recommendations if replication is required.

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.

Audit:
Run the following code snippet to determine if distribution is setup for replication (required):

```
EXECUTE sp_get_distributor;
```

Default Value:
Replication is not enabled by default.

References:
http://technet.microsoft.com/en-us/sqlserver/bb895875.aspx

1.8 Other SQL Server Configuration Options

Description:
Set the number of logs retained based on the maximum number of restarts and log cycling which may occur within your desired log retention window. The default value of 6 may be too low for many installations.

Rationale:
Keep as many SQL Server error logs as possible. The more information you have in regards to past errors, the more likely you will be able to diagnose past security issues have may have went unnoticed.

Remediation:
Consider the amount of SQL Server error log data you should retain. The default is 6 files. If you require more, adjust the setting as needed.

Audit:
The setting for Error Log retention may be set by navigating to the Management folder in SQL Server Management Studio. Right click SQL Server Logs and select Configure. The default log retention option may be altered here.

Perform the following steps to determine Error Log retention for your SQL Server instance:

1. Open SQL Server Management Studio.
2. Open Object Explorer and connect to the target SQL Server instance.
3. Expand the Management folder.
4. Right click the SQL Server Logs folder and select Configure.
5. The Maximum number of error log files may be configured in this menu.

Default Value:
By default, SQL Server will retain 6 error log files before they are recycled.

References:
http://msdn.microsoft.com/en-us/library/ms177285.aspx

1.8.1 *Database Mail (Level 1, Scorable)*

Description:
Disable Database Mail where email messaging is not required.

Rationale:
Database Mail is a new SQL Server option which allows you to easily send emails from the database engine using an SMTP server. Ensure that this service is disabled if it is not needed.

Remediation:

Determine your requirements for sending emails from the SQL engine. If messaging is not required, then ensure that Database Mail is disabled.

Audit:
Use the following code snippet to determine if Database Mail is enabled:

```
EXECUTE sp_configure 'Database Mail XPs';
```

Default Value:
Database Mail is not enabled by default.

References:
http://articles.techrepublic.com.com/5100-10878_11-6161839.html

1.8.2 *Trace Messages (Level 1, Scorable)*

Description:
Do not include execution 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 execution.

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 Table 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.
4. The 'Include execution 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

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

Description:
If a user-defined stored procedure contains sensitive propriety code 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. This should be done carefully, as encrypted stored procedures are not able to be decrypted, so the source code must be stored in the event it needs to be altered.

Remediation:
The need to encrypt stored procedure code is usually reserved for applications that are sold to the general public and contain propriety routines. If your application contains source code that you do not want to be reused by companies that purchase the application or if your application contains sensitive algorithms (such as decrypting algorithms), consider encrypting your stored procedures.

Audit:
Use the following code snippet to determine the stored procedures in a database that are encrypted.

```
SELECT
      IsEncrypted = OBJECTPROPERTY(OBJECT_ID, 'IsEncrypted'),
      ObjectName = OBJECT_NAME(OBJECT_ID)
FROM sys.sql_modules;
```

Default Value:
User-defined stored procedures are not encrypted by default.

References:
http://msdn.microsoft.com/en-us/library/aa258259(SQL.80).aspx

1.8.4 *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.

Audit:

Use the following code snippet to determine if the CLR is enabled:

```
EXECUTE sp_configure 'clr enabled'
```

Default Value:
By default, user-defined stored procedures are not extended procedures.

References:
http://support.microsoft.com/kb/190987

1.9 Extended stored procedures

The following stored procedures are disabled by default. Leave these stored procedures disable if they are not planned to be used. Do not attempt to remove any of these stored procedures. Doing so may result in an unsupported installation of SQL Server 2005.

1.9.1 `xp_availablemedia` *(Level 2, 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.

1.9.2 `xp_cmdshell` *(Level 1, Scorable)*

Description:
This stored procedure spawns a Windows command shell and passes in a string for execution.

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.

Remediation:
In most situations, users will not require executing this stored procedure. If it is absolutely required, a proxy account can be created to allow for execution 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

1.9.3 *xp_dirtree (Level 2, 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.

1.9.4 *xp_enumerrorlogs (Level 2, Scorable)*

Description:
Returns a result set of the all SQL Server error log files, containing the file number and the size of the log file.

Rationale:
Error log information should only be visible to those individuals who administer the SQL Server instance. For all other users, this information should not be retrievable. Ensure that this extended stored procedure is executable only by those individuals who require the information.

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_enumerrorlogs 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_enumerrorlogs`, 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 are not granted to the public database role.

1.9.5 *xp_enumgroups (Level 2, 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.

1.9.6 `xp_fixeddrives` *(Level 2, Scorable)*

Description:
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 is 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.

1.9.7 `xp_getnetname` *(Level 2, Scorable)*

Description:
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.

1.9.8 `xp_logevent` *(Level 2, Scorable)*

Description:
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.

1.9.9 xp_loginconfig *(Level 2, Scorable)*

Description:
Reports the login security configuration of an instance of SQL Server when it running on Windows XP, Windows Server 2003, or Windows 2000.

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

1.9.10 xp_msver *(Level 2, 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 Serve 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

1.9.11 `xp_readerrorlog` *(Level 2, 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 execution 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_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_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.

1.9.12 `xp_servicecontrol` *(Level 2, 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.

1.9.13 `xp_sprintf` *(Level 2, 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

1.9.14 `xp_sscanf` *(Level 2, 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

1.9.15 `xp_subdirs` *(Level 2, Scorable)*

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

Rationale:
This extended stored procedure gives the user executing 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.

1.10 SQL Mail extended stored procedures Description:

1.10.1 `xp_deletemail` *(Level 2, Scorable)*

Description:
This extended stored procedure deletes a message from the Microsoft SQL Server inbox. This procedure is used by `sp_processmail` to process mail in the SQL Server inbox.

Rationale:
SQL Mail stored procedures are deprecated in SQL Server 2005 and will be removed in later versions. Avoid using these stored procedures for future development purposes. Database Mail should be used as an alternative.

Remediation:
SQL Mail stored procedures can be disabled through the use of the `sp_configure` stored procedure. To ensure that SQL Mail stored procedures are disabled, perform the following steps:

```
EXECUTE sp_configure 'show advanced options',1
RECONFIGURE WITH OVERRIDE
EXECUTE sp_configure 'SQL Mail XPs',0
RECONFIGURE WITH OVERRIDE
```

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_deletemail`, right click and select Properties.
3. The permissions for this extended stored procedure can be viewed in the Permissions tab.

Default Value:
SQL Mail procedures are off by default in SQL Server 2005.

References:
http://msdn.microsoft.com/en-us/library/ms174985.aspx

1.10.2 `xp_findnextmsg` *(Level 2, Scorable)*

Description:
This extended stored procedure accepts a message ID for input and returns the message ID for output. `xp_findnextmsg` is used with `sp_processmail` in order to process mail in the Microsoft SQL Server inbox.

Rationale:
SQL Mail stored procedures are deprecated in SQL Server 2005 and will be removed in later versions. Avoid using these stored procedures for future development purposes. Database Mail should be used as an alternative.

Remediation:
SQL Mail stored procedures can be disabled through the use of the `sp_configure` stored procedure. To ensure that SQL Mail stored procedures are disabled, perform the following steps:

```
EXECUTE sp_configure 'show advanced options',1
RECONFIGURE WITH OVERRIDE
EXECUTE sp_configure 'SQL Mail XPs',0
RECONFIGURE WITH OVERRIDE
```

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_findnextmsg`, right click and select Properties.
3. The permissions for this extended stored procedure can be viewed in the Permissions tab.

Default Value:
SQL Mail procedures are off by default in SQL Server 2005.

References:
http://msdn.microsoft.com/en-us/library/aa260691%28v=SQL.80%29.aspx

1.10.3 xp_get_mapi_default_profile (Level 2, Scorable)

Description:
This extended stored procedure retrieves the default MAPI profile.

Rationale:
SQL Mail stored procedures are deprecated in SQL Server 2005 and will be removed in later versions. Avoid using these stored procedures for future development purposes. Database Mail should be used as an alternative.

Remediation:
SQL Mail stored procedures can be disabled through the use of the sp_configure stored procedure. To ensure that SQL Mail stored procedures are disabled, perform the following steps:

```
EXECUTE sp_configure 'show advanced options',1
RECONFIGURE WITH OVERRIDE
EXECUTE sp_configure 'SQL Mail XPs',0
RECONFIGURE WITH OVERRIDE
```

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_get_mapi_default_profile, right click and select Properties.
3. The permissions for this extended stored procedure can be viewed in the Permissions tab.

Default Value:
SQL Mail procedures are off by default in SQL Server 2005.

References:
No references are available for this undocumented stored procedure.

1.10.4 xp_get_mapi_profiles (Level 2, Scorable)

Description:
This extended retrieves a list of MAPI profiles for a given input.

Rationale:

SQL Mail stored procedures are deprecated in SQL Server 2005 and will be removed in later versions. Avoid using these stored procedures for future development purposes. Database Mail should be used as an alternative.

Remediation:
SQL Mail stored procedures can be disabled through the use of the `sp_configure` stored procedure. To ensure that SQL Mail stored procedures are disabled, perform the following steps:

```
EXECUTE sp_configure 'show advanced options',1
RECONFIGURE WITH OVERRIDE
EXECUTE sp_configure 'SQL Mail XPs',0
RECONFIGURE WITH OVERRIDE
```

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_getmapiprofiles`, right click and select Properties.
3. The permissions for this extended stored procedure can be viewed in the Permissions tab.

Default Value:
SQL Mail procedures are off by default in SQL Server 2005.

References:
No references are available for this undocumented stored procedure.

1.10.5 `xp_readmail` *(Level 2, Scorable)*

Description:
This extended stored procedure reads a mail message from the SQL Mail inbox. This procedure is used by `sp_processmail` to process all mail in the SQL Mail inbox.

Rationale:
SQL Mail stored procedures are deprecated in SQL Server 2005 and will be removed in later versions. Avoid using these stored procedures for future development purposes. Database Mail should be used as an alternative.

Remediation:
SQL Mail stored procedures can be disabled through the use of the `sp_configure` stored procedure. To ensure that SQL Mail stored procedures are disabled, perform the following steps:

```
EXECUTE sp_configure 'show advanced options',1
RECONFIGURE WITH OVERRIDE
EXECUTE sp_configure 'SQL Mail XPs',0
RECONFIGURE WITH OVERRIDE
```

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_readmail`, right click and select Properties.
3. The permissions for this extended stored procedure can be viewed in the Permissions tab.

Default Value:
SQL Mail procedures are off by default in SQL Server 2005.

References:
http://msdn.microsoft.com/en-us/library/aa260687%28v=SQL.80%29.aspx

1.10.6 `xp_sendmail` *(Level 2, Scorable)*

Description:
This extended stored procedure sends an e-mail message, which may include a query result set attachment, to the specified recipients. This extended stored procedure uses SQL Mail to send the message.

Rationale:
SQL Mail stored procedures are deprecated in SQL Server 2005 and will be removed in later versions. Avoid using these stored procedures for future development purposes. Database Mail should be used as an alternative.

Remediation:
SQL Mail stored procedures can be disabled through the use of the `sp_configure` stored procedure. To ensure that SQL Mail stored procedures are disabled, perform the following steps:

```
EXECUTE sp_configure 'show advanced options',1
RECONFIGURE WITH OVERRIDE
EXECUTE sp_configure 'SQL Mail XPs',0
RECONFIGURE WITH OVERRIDE
```

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_sendmail`, right click and select Properties.
3. The permissions for this extended stored procedure can be viewed in the Permissions tab.

Default Value:
SQL Mail procedures are off by default in SQL Server 2005.

References:
http://msdn.microsoft.com/en-us/library/aa260697%28v=SQL.80%29.aspx

1.10.7 `xp_startmail` *(Level 2, Scorable)*

Description:
This extended stored procedure starts a SQL Mail client session. Starting a mail session opens the `MAPI` client components and logs on to the e-mail server.

Rationale:
SQL Mail stored procedures are deprecated in SQL Server 2005 and will be removed in later versions. Avoid using these stored procedures for future development purposes. Database Mail should be used as an alternative.

Remediation:
SQL Mail stored procedures can be disabled through the use of the `sp_configure` stored procedure. To ensure that SQL Mail stored procedures are disabled, perform the following steps:

```
EXECUTE sp_configure 'show advanced options',1
RECONFIGURE WITH OVERRIDE
EXECUTE sp_configure 'SQL Mail XPs',0
RECONFIGURE WITH OVERRIDE
```

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_stopmail`, right click and select Properties.
3. The permissions for this extended stored procedure can be viewed in the Permissions tab.

Default Value:
SQL Mail procedures are off by default in SQL Server 2005.

References:
http://msdn.microsoft.com/en-us/library/aa260699%28v=SQL.80%29.aspx

1.10.8 `xp_stopmail` *(Level 2, Scorable)*

Description:
This extended stored procedure stops a SQL Mail client session. Stopping a mail session opens the `MAPI` client components and logs off of the e-mail server.

Rationale:
SQL Mail stored procedures are deprecated in SQL Server 2005 and will be removed in later versions. Avoid using these stored procedures for future development purposes. Database Mail should be used as an alternative.

Remediation:
SQL Mail stored procedures can be disabled through the use of the `sp_configure` stored procedure. To ensure that SQL Mail stored procedures are disabled, perform the following steps:

```
EXECUTE sp_configure 'show advanced options',1
RECONFIGURE WITH OVERRIDE
EXECUTE sp_configure 'SQL Mail XPs',0
RECONFIGURE WITH OVERRIDE
```

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_stopmail`, right click and select Properties.
3. The permissions for this extended stored procedure can be viewed in the Permissions tab.

Default Value:
SQL Mail procedures are off by default in SQL Server 2005.

References:
http://msdn.microsoft.com/en-us/library/aa260701%28v=SQL.80%29.aspx

1.11 WebTask extended stored procedures

Web Task stored procedures are deprecated in SQL Server 2005, so avoid using them for future development. These stored procedures generate HTML documents based on SQL Server queries or stored procedures. These procedures are disabled by default in SQL Server 2005.

1.11.1 `xp_cleanupwebtask` *(Level 2, Scorable)*

Description:

Internal stored procedure called by Enterprise Manager in SQL 2000 to clean up web task entries after their system task entry has been deleted.

Rationale:
Web assistant stored procedures are deprecated in SQL Server 2005 and will be removed in later versions. Avoid using these stored procedures for future development purposes. Reporting Services should be used as an alternative.

Remediation:
Web Assistant stored procedures can be disabled through the use of the `sp_configure` stored procedure. To ensure that Web Assistant stored procedures are disabled, perform the following steps:

```
EXECUTE sp_configure 'show advanced options',1
RECONFIGURE WITH OVERRIDE
EXECUTE sp_configure 'web assistant procedures',0
RECONFIGURE WITH OVERRIDE
```

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_cleanupwebtask`, right click and select Properties.
3. The permissions for this extended stored procedure can be viewed in the Permissions tab.

Default Value:
Web assistant procedures are off by default in SQL Server 2005.

References:
No references are available for this undocumented stored procedure.

1.11.2 `xp_convertwebtask` *(Level 2, Scorable)*

Description:
This extended stored procedure converts 6.5 webtasks to 7.0 format.

Rationale:
Web assistant stored procedures are deprecated in SQL Server 2005 and will be removed in later versions. Avoid using these stored procedures for future development purposes. Reporting Services should be used as an alternative.

Remediation:
Web Assistant stored procedures can be disabled through the use of the `sp_configure` stored procedure. To ensure that Web Assistant stored procedures are disabled, perform the following steps:

```
EXECUTE sp_configure 'show advanced options',1
RECONFIGURE WITH OVERRIDE
EXECUTE sp_configure 'web assistant procedures',0
RECONFIGURE WITH OVERRIDE
```

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_convertwebtask`, right click and select Properties.
3. The permissions for this extended stored procedure can be viewed in the Permissions tab.

Default Value:
Web assistant procedures are off by default in SQL Server 2005.

References:
No references are available for this undocumented stored procedure.

1.11.3 *xp_dropwebtask (Level 2, Scorable)*

Description:
This extended stored procedure drops a previously defined web page task.

Rationale:
Web assistant stored procedures are deprecated in SQL Server 2005 and will be removed in later versions. Avoid using these stored procedures for future development purposes. Reporting Services should be used as an alternative.

Remediation:
Web Assistant stored procedures can be disabled through the use of the `sp_configure` stored procedure. To ensure that Web Assistant stored procedures are disabled, perform the following steps:

```
EXECUTE sp_configure 'show advanced options',1
RECONFIGURE WITH OVERRIDE
EXECUTE sp_configure 'web assistant procedures',0
RECONFIGURE WITH OVERRIDE
```

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_dropwebtask`, right click and select Properties.
3. The permissions for this extended stored procedure can be viewed in the Permissions tab.

Default Value:
Web assistant procedures are off by default in SQL Server 2005.

References:
http://msdn.microsoft.com/en-us/library/aa933297%28v=SQL.80%29.aspx

1.11.4 `xp_enumcodepages` *(Level 2, Scorable)*

Description:
This extended stored procedure retrieves a list of supported code pages.

Rationale:
Web assistant stored procedures are deprecated in SQL Server 2005 and will be removed in later versions. Avoid using these stored procedures for future development purposes. Reporting Services should be used as an alternative.

Remediation:
Web Assistant stored procedures can be disabled through the use of the `sp_configure` stored procedure. To ensure that Web Assistant stored procedures are disabled, perform the following steps:

```
EXECUTE sp_configure 'show advanced options',1
RECONFIGURE WITH OVERRIDE
EXECUTE sp_configure 'web assistant procedures',0
RECONFIGURE WITH OVERRIDE
```

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_enumcodepages`, right click and select Properties.
3. The permissions for this extended stored procedure can be viewed in the Permissions tab.

Default Value:
Web assistant procedures are off by default in SQL Server 2005.

References:
No references are available for this undocumented stored procedure.

1.11.5 `xp_makewebtask` *(Level 2, Scorable)*

Description:
Creates a task that produces an HTML document containing data returned by executed queries.

Rationale:
Web assistant stored procedures are deprecated in SQL Server 2005 and will be removed in later versions. Avoid using these stored procedures for future development purposes. Reporting Services should be used as an alternative.

Remediation:
Web Assistant stored procedures can be disabled through the use of the `sp_configure` stored procedure. To ensure that Web Assistant stored procedures are disabled, perform the following steps:

```
EXECUTE sp_configure 'show advanced options',1
RECONFIGURE WITH OVERRIDE
EXECUTE sp_configure 'web assistant procedures',0
RECONFIGURE WITH OVERRIDE
```

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_makewebtask`, right click and select Properties.
3. The permissions for this extended stored procedure can be viewed in the Permissions tab.

Default Value:
Web assistant procedures are off by default in SQL Server 2005.

References:
http://msdn.microsoft.com/en-us/library/aa238843%28v=SQL.80%29.aspx

1.11.6 `xp_readwebtask` *(Level 2, Scorable)*

Description:

This extended stored procedure retrieves a listing of web task parameters.

Rationale:
Web assistant stored procedures are deprecated in SQL Server 2005 and will be removed in later versions. Avoid using these stored procedures for future development purposes. Reporting Services should be used as an alternative.

Remediation:
Web Assistant stored procedures can be disabled through the use of the `sp_configure` stored procedure. To ensure that Web Assistant stored procedures are disabled, perform the following steps:

```
EXECUTE sp_configure 'show advanced options',1
RECONFIGURE WITH OVERRIDE
EXECUTE sp_configure 'web assistant procedures',0
RECONFIGURE WITH OVERRIDE
```

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_readwebtask`, right click and select Properties.
3. The permissions for this extended stored procedure can be viewed in the Permissions tab.

Default Value:
Web assistant procedures are off by default in SQL Server 2005.

References:
No references are available for this undocumented stored procedure.

1.11.7 `xp_runwebtask` *(Level 2, Scorable)*

Description:
Executes a previously defined Web job and generates the HTML document. The task to run is identified by the output file name, by the procedure name, or by both parameters.

Rationale:
Web assistant stored procedures are deprecated in SQL Server 2005 and will be removed in later versions. Avoid using these stored procedures for future development purposes. Reporting Services should be used as an alternative.

Remediation:
Web Assistant stored procedures can be disabled through the use of the `sp_configure`

stored procedure. To ensure that Web Assistant stored procedures are disabled, perform the following steps:

```
EXECUTE sp_configure 'show advanced options',1
RECONFIGURE WITH OVERRIDE
EXECUTE sp_configure 'web assistant procedures',0
RECONFIGURE WITH OVERRIDE
```

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_runwebtask`, right click and select Properties.
3. The permissions for this extended stored procedure can be viewed in the Permissions tab.

Default Value:
Web assistant procedures are off by default in SQL Server 2005.

References:
http://msdn.microsoft.com/en-us/library/aa238890%28v=SQL.80%29.aspx

1.12 OLE Automation stored procedures

SQL Server supports the following system stored procedures that allow OLE Automation objects to be used within a Transact-SQL batch. By default, SQL Server blocks access to OLE Automation stored procedures because this component is turned off as part of the security configuration for the server.

1.12.1 `sp_OACreate` *(Level 2, Scorable)*

Description:
Execution of this procedure creates an instance of an OLE object.

Rationale:
The permissions to execute this stored procedure require membership to the `sysadmin` fixed server role.

Remediation:
`sysadmin` privileges should be reserved for only those database administrators and system administers who require the necessary permissions to perform their job functions. Assigning `sysadmin` privileges for the purpose of creating code that creates OLE objects is strongly discouraged. Remove these permissions if they are present and consider writing OLE code outside of the scope of the database engine.

Audit:
Use the following code snippet to determine which logins on the database instance have `sysadmin` privileges.

```
SELECT SysAdminMember = m.name
FROM sys.server_role_members rm
INNER JOIN sys.server_principals m ON rm.member_principal_id = m.principal_id
INNER JOIN sys.server_principals r ON rm.role_principal_id = r.principal_id
WHERE r.name = 'sysadmin'
```

Default Value:
This stored procedure is disabled by default.

References:
http://msdn.microsoft.com/en-us/library/ms189763.aspx

1.12.2 `sp_OADestroy` *(Level 2, Scorable)*

Description:
Execution of this stored procedure destroys a previously created OLE object.

Rationale:
The permissions to execute this stored procedure require membership to the `sysadmin` fixed server role.

Remediation:
`sysadmin` privileges should be reserved for only those database administrators and system administers who require the necessary permissions to perform their job functions. Assigning `sysadmin` privileges for the purpose of creating code that creates OLE objects is strongly discouraged. Remove these permissions if they are present and consider writing OLE code outside of the scope of the database engine.

Audit:
Use the following code snippet to determine which logins on the database instance have `sysadmin` privileges.

```
SELECT SysAdminMember = m.name
FROM sys.server_role_members rm
INNER JOIN sys.server_principals m ON rm.member_principal_id = m.principal_id
INNER JOIN sys.server_principals r ON rm.role_principal_id = r.principal_id
WHERE r.name = 'sysadmin';
```

Default Value:

This stored procedure is disabled by default.

References:
http://msdn.microsoft.com/en-us/library/ms173866.aspx

1.12.3 sp_OAGetErrorInfo *(Level 2, Scorable)*

Description:
Execution of this stored procedure returns OLE Automation error information.

Rationale:
The permissions to execute this stored procedure require membership to the `sysadmin` fixed server role.

Remediation:
`sysadmin` privileges should be reserved for only those database administrators and system administers who require the necessary permissions to perform their job functions. Assigning `sysadmin` privileges for the purpose of creating code that creates OLE objects is strongly discouraged. Remove these permissions if they are present and consider writing OLE code outside of the scope of the database engine.

Audit:
Use the following code snippet to determine which logins on the database instance have `sysadmin` privileges.

```
SELECT SysAdminMember = m.name
FROM sys.server_role_members rm
INNER JOIN sys.server_principals m ON rm.member_principal_id = m.principal_id
INNER JOIN sys.server_principals r ON rm.role_principal_id = r.principal_id
WHERE r.name = 'sysadmin';
```

Default Value:
This stored procedure is disabled by default.

References:
http://msdn.microsoft.com/en-us/library/ms188742.aspx

1.12.4 sp_OAGetProperty *(Level 2, Scorable)*

Description:
Execution of this stored procedure gets a property value of an OLE object.

Rationale:
The permissions to execute this stored procedure require membership to the `sysadmin` fixed server role.

Remediation:

`sysadmin` privileges should be reserved for only those database administrators and system administers who require the necessary permissions to perform their job functions. Assigning `sysadmin` privileges for the purpose of creating code that creates OLE objects is strongly discouraged. Remove these permissions if they are present and consider writing OLE code outside of the scope of the database engine.

Audit:
Use the following code snippet to determine which logins on the database instance have `sysadmin` privileges.

```
SELECT SysAdminMember = m.name
FROM sys.server_role_members rm
INNER JOIN sys.server_principals m ON rm.member_principal_id = m.principal_id
INNER JOIN sys.server_principals r ON rm.role_principal_id = r.principal_id
WHERE r.name = 'sysadmin';
```

Default Value:
This stored procedure is disabled by default.

References:
http://msdn.microsoft.com/en-us/library/ms175079.aspx

1.12.5 `sp_OAMethod` *(Level 2, Scorable)*

Description:
Execution of this stored procedure calls a method of an OLE object created by using `sp_OACreate`.

Rationale:
The permissions to execute this stored procedure require membership to the `sysadmin` fixed server role.

Remediation:
`sysadmin` privileges should be reserved for only those database administrators and system administers who require the necessary permissions to perform their job functions. Assigning `sysadmin` privileges for the purpose of creating code that creates OLE objects is strongly discouraged. Remove these permissions if they are present and consider writing OLE code outside of the scope of the database engine.

Audit:
Use the following code snippet to determine which logins on the database instance have `sysadmin` privileges.

```
SELECT SysAdminMember = m.name
FROM sys.server_role_members rm
INNER JOIN sys.server_principals m ON rm.member_principal_id = m.principal_id
INNER JOIN sys.server_principals r ON rm.role_principal_id = r.principal_id
```

```
WHERE r.name = 'sysadmin';
```

Default Value:
This stored procedure is disabled by default.

References:
http://msdn.microsoft.com/en-us/library/ms174984.aspx

1.12.6 `sp_OASetProperty` *(Level 2, Scorable)*

Description:
Execution of this stored procedure sets a property of an OLE object to a new value.

Rationale:
The permissions to execute this stored procedure require membership to the `sysadmin` fixed server role.

Remediation:
`sysadmin` privileges should be reserved for only those database administrators and system administers who require the necessary permissions to perform their job functions. Assigning `sysadmin` privileges for the purpose of creating code that creates OLE objects is strongly discouraged. Remove these permissions if they are present and consider writing OLE code outside of the scope of the database engine.

Audit:
Use the following code snippet to determine which logins on the database instance have `sysadmin` privileges.

```
SELECT SysAdminMember = m.name
FROM sys.server_role_members rm
INNER JOIN sys.server_principals m ON rm.member_principal_id = m.principal_id
INNER JOIN sys.server_principals r ON rm.role_principal_id = r.principal_id
WHERE r.name = 'sysadmin';
```

Default Value:
This stored procedure is disabled by default.

References:
http://msdn.microsoft.com/en-us/library/ms174374.aspx

1.12.7 `sp_OAStop` *(Level 2, Scorable)*

Description:
Execution of this procedure stops the server-wide OLE Automation stored procedure execution environment.

Rationale:
The permissions to execute this stored procedure require membership to the `sysadmin` fixed server role.

Remediation:
`Sysadmin` privileges should be reserved for only those database administrators and system administers who require the necessary permissions to perform their job functions. Assigning `sysadmin` privileges for the purpose of creating code that creates OLE objects is strongly discouraged. Remove these permissions if they are present and consider writing OLE code outside of the scope of the database engine.

Audit:
Use the following code snippet to determine which logins on the database instance have `sysadmin` privileges.

```
SELECT SysAdminMember = m.name
FROM sys.server_role_members rm
INNER JOIN sys.server_principals m ON rm.member_principal_id = m.principal_id
INNER JOIN sys.server_principals r ON rm.role_principal_id = r.principal_id
WHERE r.name = 'sysadmin';
```

Default Value:
This stored procedure is disabled by default.

References:
http://msdn.microsoft.com/en-us/library/ms187915.aspx

1.13 Registry access extended stored procedures

The following stored procedures are used internally by SQL Server. These procedures access and modify the system registry and should not be executed by users on the SQL Server system. To ensure that users are not able to execute these stored procedures; we will deny execute permissions to them. Do not attempt to delete any of these extended stored procedures as doing so will result in an unsupported SQL Server 2005 installation.

1.13.1 `xp_regaddmultistring` *(Level 2, Scorable)*

Description:
This extended stored procedure is used to add multiple strings to the server's registry.

Rationale:
Only on rare occasions should a string value be added to the registry from within SQL Server. When warranted, this activity should only be completed by a system administrator on the machine. Revoking this ability to general users on the SQL Server instance is necessary to ensure that the registry values are not altered. Serious problems might occur if you modify the registry incorrectly by using Registry Editor or by using another method.

These problems might require that you reinstall the operating system. Microsoft cannot guarantee that these problems can be solved.

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.

1.13.2 `xp_regdeletekey` *(Level 2, Scorable)*

Description:
This extended stored procedure is used to delete registry keys from the server's registry.

Rationale:
Only on rare occasions should a key be deleted from the registry from within SQL Server. When warranted, this activity should only be completed by a system administrator on the machine. Revoking this ability to general users on the SQL Server instance is necessary to ensure that the registry values are not altered. Serious problems might occur if you modify the registry incorrectly by using Registry Editor or by using another method. These problems might require that you reinstall the operating system. Microsoft cannot guarantee that these problems can be solved.

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.

1.13.3 `xp_regdeletevalue` *(Level 2, Scorable)*

Description:
This extended stored procedure is used to delete values from the server's registry.

Rationale:
Only on rare occasions should a value be deleted from the registry from within SQL Server. When warranted, this activity should only be completed by a system administrator on the machine. Revoking this ability to general users on the SQL Server instance is necessary to ensure that the registry values are not altered. Serious problems might occur if you modify the registry incorrectly by using Registry Editor or by using another method. These problems might require that you reinstall the operating system. Microsoft cannot guarantee that these problems can be solved.

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.

1.13.4 xp_regenumvalues *(Level 2, Scorable)*

Description:
This extended stored procedure is used to enumerate a set of values in a registry path.

Rationale:
Only on rare occasions should there be a need to enumerate a set of values from the server's registry from within SQL Server. When warranted, this activity should only be completed by a system administrator on the machine. Revoking this ability to general users on the SQL Server instance is necessary to ensure that the registry values are not altered. Serious problems might occur if you modify the registry incorrectly by using Registry Editor or by using another method. These problems might require that you reinstall the operating system. Microsoft cannot guarantee that these problems can be solved.

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.

1.13.5 xp_regremovemultistring *(Level 2, Scorable)*

Description:
This extended stored procedure is used to remove multiple strings from the server's registry.

Rationale:
Only on rare occasions should the removal of string values from the registry occur within SQL Server. When warranted, this activity should only be completed by a system administrator on the machine. Revoking this ability to general users on the SQL Server instance is necessary to ensure that the registry values are not altered. Serious problems

might occur if you modify the registry incorrectly by using Registry Editor or by using another method. These problems might require that you reinstall the operating system. Microsoft cannot guarantee that these problems can be solved.

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.

1.13.6 `xp_regwrite` *(Level 2, Scorable)*

Description:
This extended stored procedure is used to write key values to the server's registry.

Rationale:
Only on rare occasions should a value be written to the registry from within SQL Server. When warranted, this activity should only be completed by a system administrator on the machine. Revoking this ability to general users on the SQL Server instance is necessary to ensure that the registry values are not altered. Serious problems might occur if you modify the registry incorrectly by using Registry Editor or by using another method. These problems might require that you reinstall the operating system. Microsoft cannot guarantee that these problems can be solved.

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 permissions are not revoked on this system stored procedure by default.

References:
This stored procedure is undocumented and thus no official reference exists.

1.13.7 *SQL Server Event Forwarding (Level 1, 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 for 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

1.13.8 *SQL Server Browser Service (Level 1, Scorable)*

Description:
A windows service that listens for incoming requests for SQL Server resources and provides information about SQL Server instances installed on the computer.

Rationale:
Disabling this service leads to a move secure SQL Server installation because port numbers are required to be specified when connecting to the SQL Server instance.

Remediation:
When the SQL Browser service is disabled, all connections to named instances must provide the TCP/IP port or the named pipe. Also, the dedicated administrator connection (DAC) will not work if the default instance is not using port 1433.

Audit:
Perform the following steps to determine the run status of the SQL Server Browser Service:

1. Run `services.msc` from the Run menu.
2. Locate SQL Server Browser from the services list.
3. Right click this service and select Properties.

Default Value:
The SQL Browser service is set to start automatically by default.

References:
http://msdn.microsoft.com/en-us/library/ms181087.aspx

1.14 Authentication and Authorization

1.14.1 *SQL Server install directory permissions (Level 1, Scorable)*

Description:
Modify the permissions to the `%ProgramFiles%\Microsoft SQL Server` directory. Assign the SQL Server service account Full Control. Remove the Users group's permission. The SYSTEM, Administrators, and TrustedInstaller groups should remain.

Remediation:
The SQL Server tools are housed in the Program Files\Microsoft SQL Server directory. Secure the permissions on this folder to disallow any potential malicious activity to the SQL Server binaries and tools.

Audit:
Navigate to the `%ProgramFiles%\Microsoft SQL Server` directory to review the folder permissions.

Default Value:
By default, the system administrator, power users, and current user will have permissions to this folder.

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

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

Description:

Delete or secure old setup files. Protect files in the `%ProgramFiles%\Microsoft SQL Server\MSSQL.X\MSSQL\Install`, e.g., `sqlstp.log`, `sqlsp.log` and `setup.iss`. *X* represents the installations of various SQL Server installs due to the fact that multiple instances of SQL Server or SQL Express can be installed.

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:
The SQL Server tools are housed in the `%ProgramFiles%\Microsoft SQL Server\MSSQL.X\MSSQL\Install`. Secure the permissions on this folder to disallow any potential malicious activity to the SQL Server binaries and tools.

Audit:
Navigate to the `%ProgramFiles%\Microsoft SQL Server\MSSQL.X\MSSQL\Install` directory to review the folder permissions.

Default Value:
By default, the system administrator, power users, and current user will have permissions to this folder.

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

1.14.3 *Assigning System Administrators role (Level 1, Not Scorable)*

Description:
When assigning database administrators to the System Administrators role, map their Windows accounts to SQL logins, and then assign them to the role. Assign only authorized DBAs to the SQL Server System Administrators role.

Rationale:
Only those individuals who you expect to perform administrative tasks should have assigned to the `sysadmin` (sa) system role.

Remediation:
Revoke `sysadmin` privileges to any account that does not perform administrative duties that require it.

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, all members of the Windows BUILTIN\Administrators group, the local administrator's group, are members of the `sysadmin` fixed server role.

References:
http://msdn.microsoft.com/en-us/library/ms178032%28SQL.90%29.aspx

1.14.4 *SQL Logins (Level 1, Scorable)*

Description:
Remove the default `BUILTIN\Administrators` windows login.

Rationale:
The `BUILTIN\Administrators` group login is a member of the `sysadmin` server role by default, which allows system administrators on the operating system `sysadmin` rights to the database. Removing this login removes `sysadmin` rights to the system administrators on the machine. Do not remove `BUILTIN\Administrators` until another account has been assigned the System Administrators role.

Remediation:
Remove the `BUILTIN\Administrators` account from the `sysadmin` server role. If the account is required, ensure that no unauthorized users are assigned to this role at the OS level.

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, the `BUILTIN\Administrators` group is a member of the `sysadmin` login on the SQL instance.

References:
http://msdn.microsoft.com/en-us/library/ms178032%28SQL.90%29.aspx

1.14.5 *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. If the host's Operating System is Windows Server 2003 or higher, 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:
If on Windows Server 2003 or higher, ensure that the `CHECK_POLICY` option is enabled for all SQL logins. If not on Windows Server 2003 or higher, define a custom password policy and ensure it is followed.

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

1.14.6 *OS Guests access (Level 1, Scorable)*

Description:
Deny database login for the Guests OS group.

Rationale:
Deny any OS guest account the right to log into the SQL Server instance. Guests on the OS should not be granted access to any data in the database.

Remediation:
Review the windows logins. If the OS guest account or any similar account is a valid login, deny the login from the instance.

Audit:
Run the following code snippet to determine which windows accounts have logins on the SQL Server instance.

```
SELECT WindowsLoginName = name, IsDisabled = is_disabled
FROM sys.server_principals
WHERE type_desc = 'WINDOWS_LOGIN';
```

To deny the login:

```
EXECUTE sp_denylogin 'Computer_Name\Guests';
```

Default Value:
The OS Guest group is not a SQL login by default.

References:
http://technet.microsoft.com/en-us/library/cc966454.aspx

1.14.7 *Fixed Server Roles (Level 1, Not Scorable)*

Description:
Only use the fixed server roles sysadmin, 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 *SQL Server Database Users and Roles (Level 1, Scorable)*

Description:
The guest account allows 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 from all databases except master and tempdb.

Rationale:
The guest account cannot be revoked `CONNECT` permissions from master and tempdb, but should be revoked from 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/aa905195(SQL.80).aspx

1.14.9 *Statement 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 and schema owners, not individual users. Individual users are those that consume the data, not the users that alter the database schema.

Remediation:
Review your DDL permission scheme. Consider revoking DDL permissions from the users who do not frequently execute DDL statements. 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

1.14.10 *Low-privileged users (Level 1, Scorable)*

Description:
Do not grant object permissions to `PUBLIC` or `GUEST`.

Rationale:
Public is a database role in every database. When a user has not been granted or denied specific permissions on a securable the user will inherit permissions granted to the public role.

Guest is a user account in every database. Permissions granted to the guest user account are inherited by users who do not have an account in the database.

Remediation:
Ensuring that the public database role and the guest user account do not have explicit permissions is a great first step in securing your database. The Public database role should be granted only minimal permissions to read data and no permissions to modify data. The guest user account should have the `CONNECT` permissions revoked so that only explicit users can connect to the database.

Audit:

Use the following code snippet to determine the specific permissions given to the Public database user role:

```
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
      p.name = 'public';
```

Default Value:
By default, every database user is a member of the public role. A user that attempts to connect to a database without explicit permissions assume the permissions of the guest database user.

References:
http://msdn.microsoft.com/en-us/library/ms181127.aspx

1.14.11 *Stored Procedure Permissions (Level 1, Scorable)*

Description:
Grant executes permissions on stored procedures to database roles (not users).

Rationale:
Assigning `EXECUTE` permissions to database roles rather than individual user accounts greatly ease security administration.

Remediation:
Perform a security audit to determine which user accounts have explicit stored procedure permissions. Consider moving these users to roles and assign execute permissions to the roles.

Audit:
Run the following code snippet to determine which users have explicit permissions for stored procedures.

```
SELECT state_desc, permission_name, ProcedureName = p.name, prin.type_desc,
prin.name
FROM sys.database_permissions dp
JOIN sys.procedures p on dp.major_id = p.object_id
JOIN sys.database_principals prin ON dp.grantee_principal_id =
prin.principal_id;
```

Default Value:
Users are not given explicit permissions to run stored procedures by default. The schema owner and database owner have permissions to execute procedures in the database.

References:
http://msdn.microsoft.com/en-us/library/ms345484.aspx

1.14.12 *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, keep an eye on their activities.

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.aspx

1.14.13 *SQL Server Agent subsystem privileges (Level 1, Not Scorable)*

Description:
Restrict the SQL Server Agent proxy access to required/approved subsystems.

Rationale:
Allowing access to CmdExec and ActiveX subsystems allows direct OS access and should be avoided unless business justifications for doing so exist.

Remediation:
Consider removing elevated and unneeded permissions from the SQL Server Agent proxy account.

Audit:
Review all SQL Server and OS permissions for the current SQL Server proxy accounts. To view the SQL Server permissions for a given database, run the following script:

```
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;
```

Run the following script to determine server-level permissions:

```
SELECT dp.permission_name, dp.state_desc, p.name, p.type_desc
FROM sys.server_permissions dp
join sys.server_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 the SQL Server agent proxy is not given elevated permissions.

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

1.14.14 *User-defined Database Roles (Level 1, 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:
http://msdn.microsoft.com/en-us/library/aa905188(SQL.80).aspx

1.14.15 *Fixed Database Roles (Level 1, Scorable)*

Description:

Only use the fixed database role `db_owner` to support DBA activity.

Rationale:
Only database administrators should have elevated permissions. Application accounts, developers, and other non-db 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 perform regular administrative activity.

Audit:
Run the following code snippet to determine server role members:

```
SELECT 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;
```

Default Value:
New users are not members of the `db_owner` role by default.

References:
http://msdn.microsoft.com/en-us/library/ms180977%28SQL.90%29.aspx

1.14.16 *Users and Roles (Level 1, Scorable)*

Description:
Ensure that the members of the roles (users/groups/other roles) in the target database actually exist.

Rationale:

Remove all unused accounts. If an account is no longer used it should be removed to ensure it is not compromised by an attacked and used maliciously.

Remediation:
Review all user and groups in all production databases as well as all server logins. Remove any accounts that are not longer used.

Audit:
Use the following code snippet to determine the database users and roles in a particular database:

```
SELECT name, type_desc
FROM sys.database_principals;
```

The following code snippet can be used to find all logins and roles in a particular instance:

```
SELECT name, type_desc
FROM sys.server_principals;
```

Default Value:
By default, a user is a member of the `public` database role and `guest` login.

References:
http://msdn.microsoft.com/en-us/library/ms188283.aspx

1.14.17 *Application Roles (Level 1, Not Scorable)*

Description:
Use application roles to limit access to data to users of specific applications. Use encryption to protect the role name and password in the connection string. Use `EXECUTE AS WITH NO REVERT` or `WITH COOKIE` to allow individuals to access the application without knowing the password.

Rationale:
This provides permission based rather than password based mechanism to sandbox access. An application role is a database principal that enables an application to run with its own, user-like permissions. You can use application roles to enable access to specific data to only those users who connect through a particular application. Unlike database roles, application roles contain no members and are inactive by default. Application roles work with both authentication modes.

Remediation:
Consider using application roles for those situations where the database users ONLY connect through the application. In other situations, application roles may not make sense as there is additional overhead involved in maintaining the security of the application role and the other use base.

Audit:
Use the following code snippet to find any existing application roles in a given database.

```
SELECT *
FROM sys.database_principals
WHERE type = 'A';
```

Default Value:
Application roles are not used by default.

References:
http://msdn.microsoft.com/en-us/library/ms190998.aspx

1.14.18 *Use of Predefined Database Roles (Level 1, Scorable)*

Description:
Avoid assigning the Guest user to any predefined database role.

Rationale:
Predefined database roles are designed for specific activity, much of it administrative activity on the database. Do not assign any of these roles to the Guest user.

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 relationship between database users and database roles.

```
SELECT
      RoleName = r.name,
      UserName = p.name
FROM sys.database_role_members rm
JOIN sys.database_principals p on rm.member_principal_id = p.principal_id
JOIN sys.database_principals r on rm.role_principal_id = r.principal_id;
```

Default Value:
By default, a user is only a member of the public database role. A user must be explicitly added to a database role to be a member.

References:
http://msdn.microsoft.com/en-us/library/aa905188(SQL.80).aspx

1.14.19 *Do not use Remote Servers (Level 1, Scorable)*

Description:

Use linked servers rather than remote servers where required. Remove any unused linked servers or disable this feature.

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. Remove (right-click the linked server and select delete fro

Remediation:
Evaluate all linked servers on the instance. Consider removing any linked servers that are not actively used.

Audit:
Use the following code snippet to find all linked servers on the instance.

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

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

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

1.14.20 *Linked or Remote Servers (Level 1, Scorable)*

Description:
Configure linked or remote servers to use Windows authentication where required. Disable linked servers otherwise. 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.aspx

1.14.21 *Linked Server logins (Level 1, Not Scorable)*

Description:
Allow linked server access only to those logins that need it. Disable linked servers otherwise.

Rationale:
When possible, configure linked servers to connect under the current user's security context.

Remediation:
Evaluate the current linked servers on the system. When possible, ensure the linked server authentication occurs under the current user's security context.

Audit:
Use the following code snippet to identify all linked server logins:

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

Default Value:
Linked servers are not defined by default unless replication is defined.

References:
http://msdn.microsoft.com/en-us/library/aa213778(SQL.80).aspx

1.14.22 *Ad Hoc Distributed Queries (Level 1, Not Scorable)*

Description:
Disable ad hoc distributed query access on all providers for all users except members of the `sysadmin` fixed role.

Rationale:
If SQL Authentication is used, Ad Hoc data access increases the probability of credential disclosure via the practice of embedding username and password in the provider string.

Remediation:
Ensure the ad-hoc distributed queries option is disabled when not required.

Audit:

Use the following code snippet to identify the status of the ad-doc distributed queries option:

```
sp_configure 'show advanced options', 1
RECONFIGURE
sp_configure 'Ad Hoc Distributed Queries'
```

Default Value:
Ad hoc distributed queries are disabled by default.

References:
http://msdn.microsoft.com/en-us/library/ms187569.aspx

1.15 Auditing and Logging

1.15.1 *Auditing – General (Level 1, Not Scorable)*

Description:
Prepare a schedule for reviewing audit information 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:
http://msdn.microsoft.com/en-us/library/cc297240.aspx

1.15.2 *SQL Server Properties – Security Tab (Level 1, Scorable)*

Description:
Through the SQL Server Management Studio, enable auditing for SQL Server.

Rationale:
At a minimum, enable failed login attempts. Auditing of failed login attempts only is enabled by default.

Remediation:
Consider capturing failed and successful login attempts. Failed login attempts should always be audited. These events are captured in the Event Log.

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. This auditing property is set under the "Login Auditing" section.

Default Value:
By default, failed login attempted are captured.

References:
http://msdn.microsoft.com/en-us/library/ms175850.aspx

1.15.3 *SQL Server Logs (Level 1, Not Scorable)*

Description:
SQL Server audit data must be protected from loss. The SQL Server and SQL Server Agent logs must be backed up before they are overwritten.

Rationale:
Adjust the number of logs to prevent data loss. The default is six. This may be too low for your production environment. Consider increasing the default number of SQL Logs from the default of 6.

Remediation:
Consider the amount of SQL Server log information you should keep. Adjust as necessary.

Audit:
Perform the following steps to adjust SQL Server Error Log retention:

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.

Default Value:
The default number of SQL Server logs kept is 6. After 6 files have been reached, recycling occurs.

References:
http://msdn.microsoft.com/en-us/library/ms191202.aspx

1.15.4 *SQL Profiler (Level 1, Scorable)*

Description:

SQL Server Profiler may be used to identify system events that may indicate security risks.

Rationale:
SQL Server Profiler is a user application that captures events that occur on the SQL Server system through the use of defined traces. There may significant amount of overhead through running SQL Profiler depending on where the application is ran and how many events are captured. It is a great tool for capturing data over finite periods of time, but is not recommended as a long-term solution for continually capturing system events.

Remediation:
Consider using SQL Server Profiler to capture security related system events if you are not already doing so. This tool is very useful in pinpointing security threats as they are happening.

Audit:
SQL Server Profiler ships with all versions of SQL Server except SQL Server Express Edition.

Default Value:
When SQL Server Profiler is started a default trace is used. These traces may be altered or you may define and save your own trace templates detailing the security events you want to capture when the trace is ran.

References:
http://msdn.microsoft.com/en-us/library/cc297240.aspx

1.15.5 *Profiler Events*

The following are security related events that may be captured in SQL Server Profiler or via a server-side trace. Use these events to identify specific security related issues that may be occurring on your server.

1.15.6 *Audit Add DB User Event (Level 1, Scorable)*

Description:
SQL Server event occurring when a database user has been added or removed.

Rationale:
Use this trace event to determine when a database user has been added or removed. An example usage of this trace event would be when you suspect unauthorized users are creating databases on your SQL Server instance.

Remediation:
Modify your trace to add this event if it contains information you are interested in monitoring. To modify your SQL Server Profiler trace, perform the following steps:

1. Stop your trace if it is currently running. Go to File menu and choose Properties.

2. Choose the Events Selection tab on the Properties screen. Choose the Show All Events dialog box.
3. Choose the SQL Trace events that you wish to capture in your SQL Server profiler Trace.
4. Choose the Show All Columns Dialog box. Add additional data columns as necessary.

Audit:
View the server trace definition to verify that this event is being captured. The trace definition can be viewed from the SQL Server Profiler trace Properties page. To view the trace properties in SQL Server Profiler, go to the File menu and choose Properties. Choose the Events Selection table on the properties screen to view the events and columns currently being captured.

Default Value:
This event is not captured by default.

References:
http://msdn.microsoft.com/en-us/library/ms175481.aspx

1.15.7 *Audit Add Login to Server Role (Level 1, Scorable)*

Description:
SQL Server event occurring when a login has been added to or removed from a server role.

Rationale:
Use this trace event to determine when a login is assigned to or revoked from a server role. An example usage of this trace event would be when you suspect unauthorized users are being added to server roles on your SQL Server instance.

Remediation:
Modify your trace to add this event if it contains information you are interested in monitoring. To modify your SQL Server Profiler trace, perform the following steps:

1. Stop your trace if it is currently running. Go to File menu and choose Properties.
2. Choose the Events Selection tab on the Properties screen. Choose the Show All Events dialog box.
3. Choose the SQL Trace events that you wish to capture in your SQL Server profiler Trace.
4. Choose the Show All Columns Dialog box. Add additional data columns as necessary.

Audit:
View the server trace definition to verify that this event is being captured. The trace definition can be viewed from the SQL Server Profiler trace Properties page. To view the trace properties in SQL Server Profiler, go to the File menu and choose Properties. Choose

the Events Selection table on the properties screen to view the events and columns currently being captured.

Default Value:
This event is not captured by default.

References:
http://msdn.microsoft.com/en-us/library/ms175481.aspx

1.15.8 *Audit Add Member to DB Role (Level 1, Scorable)*

Description:
SQL Server event occurring when a database user is added to a database role.

Rationale:
Use this trace event to determine when a database user is added to a database role. An example usage of this trace event would be when you suspect unauthorized users are being added to a database role on your SQL Server instance.

Remediation:
Modify your trace to add this event if it contains information you are interested in monitoring. To modify your SQL Server Profiler trace, perform the following steps:

1. Stop your trace if it is currently running. Go to File menu and choose Properties.
2. Choose the Events Selection tab on the Properties screen. Choose the Show All Events dialog box.
3. Choose the SQL Trace events that you wish to capture in your SQL Server profiler Trace.
4. Choose the Show All Columns Dialog box. Add additional data columns as necessary.

Audit:
View the server trace definition to verify that this event is being captured. The trace definition can be viewed from the SQL Server Profiler trace Properties page. To view the trace properties in SQL Server Profiler, go to the File menu and choose Properties. Choose the Events Selection table on the properties screen to view the events and columns currently being captured.

Default Value:
This event is not captured by default.

References:
http://msdn.microsoft.com/en-us/library/ms175481.aspx

1.15.9 *Audit Add Role Event (Level 1, Scorable)*

Description:
SQL Server event occurring when a database role is added or removed.

Rationale:
Use this trace event to determine when a database role is added or removed. An example usage of this trace event would be when you suspect unauthorized users are adding database roles to databases on your SQL Server instance.

Remediation:
Modify your trace to add this event if it contains information you are interested in monitoring. To modify your SQL Server Profiler trace, perform the following steps:

1. Stop your trace if it is currently running. Go to File menu and choose Properties.
2. Choose the Events Selection tab on the Properties screen. Choose the Show All Events dialog box.
3. Choose the SQL Trace events that you wish to capture in your SQL Server profiler Trace.
4. Choose the Show All Columns Dialog box. Add additional data columns as necessary.

Audit:
View the server trace definition to verify that this event is being captured. The trace definition can be viewed from the SQL Server Profiler trace Properties page. To view the trace properties in SQL Server Profiler, go to the File menu and choose Properties. Choose the Events Selection table on the properties screen to view the events and columns currently being captured.

Default Value:
This event is not captured by default.

References:
http://msdn.microsoft.com/en-us/library/ms175481.aspx

1.15.10 *Audit Addlogin Event (Level 1, Scorable)*

Description:
SQL Server event occurring when a SQL Server login is added or removed.

Rationale:
Use this trace event to determine when a SQL Server login is created or dropped. An example usage of this trace event would be when you suspect unauthorized users are adding logins on your SQL Server instance.

Remediation:
Modify your trace to add this event if it contains information you are interested in monitoring. To modify your SQL Server Profiler trace, perform the following steps:

1. Stop your trace if it is currently running. Go to File menu and choose Properties.
2. Choose the Events Selection tab on the Properties screen. Choose the Show All Events dialog box.
3. Choose the SQL Trace events that you wish to capture in your SQL Server profiler Trace.
4. Choose the Show All Columns Dialog box. Add additional data columns as necessary.

Audit:
View the server trace definition to verify that this event is being captured. The trace definition can be viewed from the SQL Server Profiler trace Properties page. To view the trace properties in SQL Server Profiler, go to the File menu and choose Properties. Choose the Events Selection table on the properties screen to view the events and columns currently being captured.

Default Value:
This event is not captured by default.

References:
http://msdn.microsoft.com/en-us/library/ms175481.aspx

1.15.11 *Audit App Role Change Password (Level 1, Scorable)*

Description:
SQL Server event occurring when passwords are changed for an application role.

Rationale:
Use this trace event to determine when an application role's password is changed. An example usage of this trace event would be when you suspect unauthorized users are changing application role passwords on your SQL Server instance.

Remediation:
Modify your trace to add this event if it contains information you are interested in monitoring. To modify your SQL Server Profiler trace, perform the following steps:

1. Stop your trace if it is currently running. Go to File menu and choose Properties.
2. Choose the Events Selection tab on the Properties screen. Choose the Show All Events dialog box.

3. Choose the SQL Trace events that you wish to capture in your SQL Server profiler Trace.
4. Choose the Show All Columns Dialog box. Add additional data columns as necessary.

Audit:
View the server trace definition to verify that this event is being captured. The trace definition can be viewed from the SQL Server Profiler trace Properties page. To view the trace properties in SQL Server Profiler, go to the File menu and choose Properties. Choose the Events Selection table on the properties screen to view the events and columns currently being captured.

Default Value:
This event is not captured by default.

References:
http://msdn.microsoft.com/en-us/library/ms175481.aspx

1.15.12 *Audit Backup/Restore (Level 1, Scorable)*

Description:
SQL Server event occurring when a backup or restore statement has been issued.

Rationale:
Use this trace event to determine when a backup or restore statement has been issued. An example usage of this trace event would be when you suspect unauthorized users are create or restoring database backups on your SQL Server instance. If you've sensitive data, this is a very important consideration.

Remediation:
Modify your trace to add this event if it contains information you are interested in monitoring. To modify your SQL Server Profiler trace, perform the following steps:

1. Stop your trace if it is currently running. Go to File menu and choose Properties.
2. Choose the Events Selection tab on the Properties screen. Choose the Show All Events dialog box.
3. Choose the SQL Trace events that you wish to capture in your SQL Server profiler Trace.
4. Choose the Show All Columns Dialog box. Add additional data columns as necessary.

Audit:
View the server trace definition to verify that this event is being captured. The trace definition can be viewed from the SQL Server Profiler trace Properties page. To view the trace properties in SQL Server Profiler, go to the File menu and choose Properties. Choose

the Events Selection table on the properties screen to view the events and columns currently being captured.

Default Value:
This event is not captured by default.

References:
http://msdn.microsoft.com/en-us/library/ms175481.aspx

1.15.13 *Audit Broker Conversation (Level 1, Scorable)*

Description:
SQL Server event which reports audit messages related to Service Broker dialog security.

Rationale:
Use this trace event to determine when Service Broker dialog security events occur.

Remediation:
Modify your trace to add this event if it contains information you are interested in monitoring. To modify your SQL Server Profiler trace, perform the following steps:

1. Stop your trace if it is currently running. Go to File menu and choose Properties.
2. Choose the Events Selection tab on the Properties screen. Choose the Show All Events dialog box.
3. Choose the SQL Trace events that you wish to capture in your SQL Server profiler Trace.
4. Choose the Show All Columns Dialog box. Add additional data columns as necessary.

Audit:
View the server trace definition to verify that this event is being captured. The trace definition can be viewed from the SQL Server Profiler trace Properties page. To view the trace properties in SQL Server Profiler, go to the File menu and choose Properties. Choose the Events Selection table on the properties screen to view the events and columns currently being captured.

Default Value:
This event is not captured by default.

References:
http://msdn.microsoft.com/en-us/library/ms175481.aspx

1.15.14 *Audit Broker Login (Level 1, Scorable)*

Description:
SQL Server event which reports audit messages related to Service Broker transport

security.

Rationale:
Use this trace event to determine when a Service Broker transport security event occurs.

Remediation:
Modify your trace to add this event if it contains information you are interested in monitoring. To modify your SQL Server Profiler trace, perform the following steps:

1. Stop your trace if it is currently running. Go to File menu and choose Properties.
2. Choose the Events Selection tab on the Properties screen. Choose the Show All Events dialog box.
3. Choose the SQL Trace events that you wish to capture in your SQL Server profiler Trace.
4. Choose the Show All Columns Dialog box. Add additional data columns as necessary.

Audit:
View the server trace definition to verify that this event is being captured. The trace definition can be viewed from the SQL Server Profiler trace Properties page. To view the trace properties in SQL Server Profiler, go to the File menu and choose Properties. Choose the Events Selection table on the properties screen to view the events and columns currently being captured.

Default Value:
This event is not captured by default.

References:
http://msdn.microsoft.com/en-us/library/ms175481.aspx

1.15.15 *Audit Change Audit (Level 1, Scorable)*

Description:
SQL Server event indicating that an audit trace modification has been made.

Rationale:
Use this trace event to determine when a trace modification has been made. An example usage of this trace event would be when you suspect unauthorized users are modifying system traces on your SQL Server instance.

Remediation:
Modify your trace to add this event if it contains information you are interested in monitoring. To modify your SQL Server Profiler trace, perform the following steps:

1. Stop your trace if it is currently running. Go to File menu and choose Properties.
2. Choose the Events Selection tab on the Properties screen. Choose the Show All Events dialog box.
3. Choose the SQL Trace events that you wish to capture in your SQL Server profiler Trace.
4. Choose the Show All Columns Dialog box. Add additional data columns as necessary.

Audit:
View the server trace definition to verify that this event is being captured. The trace definition can be viewed from the SQL Server Profiler trace Properties page. To view the trace properties in SQL Server Profiler, go to the File menu and choose Properties. Choose the Events Selection table on the properties screen to view the events and columns currently being captured.

Default Value:
This event is not captured by default.

References:
http://msdn.microsoft.com/en-us/library/ms175481.aspx

1.15.16 *Audit Change Database Owner (Level 1, Scorable)*

Description:
SQL Server event indicating that permissions to change the owner of the database have been checked.

Rationale:
Use this trace event to determine when permissions to change the owner of a database have been checked. An example usage of this trace event would be when you suspect unauthorized users have changed a database owner.

Remediation:
Modify your trace to add this event if it contains information you are interested in monitoring. To modify your SQL Server Profiler trace, perform the following steps:

1. Stop your trace if it is currently running. Go to File menu and choose Properties.
2. Choose the Events Selection tab on the Properties screen. Choose the Show All Events dialog box.
3. Choose the SQL Trace events that you wish to capture in your SQL Server profiler Trace.
4. Choose the Show All Columns Dialog box. Add additional data columns as necessary.

Audit:
View the server trace definition to verify that this event is being captured. The trace definition can be viewed from the SQL Server Profiler trace Properties page. To view the trace properties in SQL Server Profiler, go to the File menu and choose Properties. Choose the Events Selection table on the properties screen to view the events and columns currently being captured.

Default Value:
This event is not captured by default.

References:
http://msdn.microsoft.com/en-us/library/ms175481.aspx

1.15.17 *Audit Database Scope GDR (Level 1, Scorable)*

Description:
SQL Server event indicating that a `GRANT`, `DENY`, or `REVOKE` event for a database object has occurred.

Rationale:
Use this trace event to determine when a `GRANT`, `DENY`, or `REVOKE` event occurs for a database object. An example usage of this trace event would be when you suspect unauthorized users are modifying database object permissions.

Remediation:
Modify your trace to add this event if it contains information you are interested in monitoring. To modify your SQL Server Profiler trace, perform the following steps:

1. Stop your trace if it is currently running. Go to File menu and choose Properties.
2. Choose the Events Selection tab on the Properties screen. Choose the Show All Events dialog box.
3. Choose the SQL Trace events that you wish to capture in your SQL Server profiler Trace.
4. Choose the Show All Columns Dialog box. Add additional data columns as necessary.

Audit:
View the server trace definition to verify that this event is being captured. The trace definition can be viewed from the SQL Server Profiler trace Properties page. To view the trace properties in SQL Server Profiler, go to the File menu and choose Properties. Choose the Events Selection table on the properties screen to view the events and columns currently being captured.

Default Value:
This event is not captured by default.

References:
http://msdn.microsoft.com/en-us/library/ms175481.aspx

1.15.18 *Audit DBCC*

Description:
SQL Server event indicating that a `GRANT`, `DENY`, or `REVOKE` event for a database object has occurred.

Rationale:
Use this trace event to determine when a `GRANT`, `DENY`, or `REVOKE` event occurs for a database object. An example usage of this trace event would be when you suspect unauthorized users are executing DBCC statements.

Remediation:
Modify your trace to add this event if it contains information you are interested in monitoring. To modify your SQL Server Profiler trace, perform the following steps:

1. Stop your trace if it is currently running. Go to File menu and choose Properties.
2. Choose the Events Selection tab on the Properties screen. Choose the Show All Events dialog box.
3. Choose the SQL Trace events that you wish to capture in your SQL Server profiler Trace.
4. Choose the Show All Columns Dialog box. Add additional data columns as necessary.

Audit:
View the server trace definition to verify that this event is being captured. The trace definition can be viewed from the SQL Server Profiler trace Properties page. To view the trace properties in SQL Server Profiler, go to the File menu and choose Properties. Choose the Events Selection table on the properties screen to view the events and columns currently being captured.

Default Value:
This event is not captured by default.

1.15.19 *Audit Database Management (Level 1, Scorable)*

Description:

SQL Server event indicating that a database has been created altered or dropped.

Rationale:
Use this event to determine when a database has been created, altered, or dropped. An example usage of this trace event would be when you suspect unauthorized users are modifying database object permissions.

Remediation:
Modify your trace to add this event if it contains information you are interested in monitoring. To modify your SQL Server Profiler trace, perform the following steps:

1. Stop your trace if it is currently running. Go to File menu and choose Properties.
2. Choose the Events Selection tab on the Properties screen. Choose the Show All Events dialog box.
3. Choose the SQL Trace events that you wish to capture in your SQL Server profiler Trace.
4. Choose the Show All Columns Dialog box. Add additional data columns as necessary.

Audit:
View the server trace definition to verify that this event is being captured. The trace definition can be viewed from the SQL Server Profiler trace Properties page. To view the trace properties in SQL Server Profiler, go to the File menu and choose Properties. Choose the Events Selection table on the properties screen to view the events and columns currently being captured.

Default Value:
This event is not captured by default.

References:
http://msdn.microsoft.com/en-us/library/ms175481.aspx

1.15.20 *Audit Database Object Access (Level 1, Scorable)*

Description:
SQL Server event indicating that a database object, such as a schema, has been accessed.

Rationale:
Use this trace event to determine when a database object, such as a schema, has been accessed.

Remediation:
Modify your trace to add this event if it contains information you are interested in monitoring. To modify your SQL Server Profiler trace, perform the following steps:

1. Stop your trace if it is currently running. Go to File menu and choose Properties.
2. Choose the Events Selection tab on the Properties screen. Choose the Show All Events dialog box.
3. Choose the SQL Trace events that you wish to capture in your SQL Server profiler Trace.
4. Choose the Show All Columns Dialog box. Add additional data columns as necessary.

Audit:
View the server trace definition to verify that this event is being captured. The trace definition can be viewed from the SQL Server Profiler trace Properties page. To view the trace properties in SQL Server Profiler, go to the File menu and choose Properties. Choose the Events Selection table on the properties screen to view the events and columns currently being captured.

Default Value:
This event is not captured by default.

References:
http://msdn.microsoft.com/en-us/library/ms175481.aspx

1.15.21 *Audit Database Object GDR (Level 1, Scorable)*

Description:
SQL Server event indicating when a `GRANT`, `REVOKE`, or `DENY` has been issued for database objects, such as assemblies and schemas.

Rationale:
Use this trace event to determine when a `GRANT`, `DENY`, or `REVOKE` command is used against a database object. An example usage of this trace event would be when you suspect unauthorized users are modifying database object permissions.

Remediation:
Modify your trace to add this event if it contains information you are interested in monitoring. To modify your SQL Server Profiler trace, perform the following steps:

1. Stop your trace if it is currently running. Go to File menu and choose Properties.
2. Choose the Events Selection tab on the Properties screen. Choose the Show All Events dialog box.
3. Choose the SQL Trace events that you wish to capture in your SQL Server profiler Trace.
4. Choose the Show All Columns Dialog box. Add additional data columns as necessary.

Audit:
View the server trace definition to verify that this event is being captured. The trace definition can be viewed from the SQL Server Profiler trace Properties page. To view the trace properties in SQL Server Profiler, go to the File menu and choose Properties. Choose the Events Selection table on the properties screen to view the events and columns currently being captured.

Default Value:
This event is not captured by default.

References:
http://msdn.microsoft.com/en-us/library/ms175481.aspx

1.15.22 *Audit Database Object Management (Level 1, Scorable)*

Description:
SQL Server event indicating when a `CREATE`, `ALTER`, or `DROP` statement is executed on database objects, such as schema objects.

Rationale:
Use this trace event to determine when a database object is created or altered. An example usage of this trace event would be when you suspect unauthorized users are creating database tables.

Remediation:
Modify your trace to add this event if it contains information you are interested in monitoring. To modify your SQL Server Profiler trace, perform the following steps:

1. Stop your trace if it is currently running. Go to File menu and choose Properties.
2. Choose the Events Selection tab on the Properties screen. Choose the Show All Events dialog box.
3. Choose the SQL Trace events that you wish to capture in your SQL Server profiler Trace.
4. Choose the Show All Columns Dialog box. Add additional data columns as necessary.

Audit:
View the server trace definition to verify that this event is being captured. The trace definition can be viewed from the SQL Server Profiler trace Properties page. To view the trace properties in SQL Server Profiler, go to the File menu and choose Properties. Choose the Events Selection table on the properties screen to view the events and columns currently being captured.

Default Value:

This event is not captured by default.

References:
http://msdn.microsoft.com/en-us/library/ms175481.aspx

1.15.23 *Audit Database Object Take Ownership (Level 1, Scorable)*

Description:
SQL Server event indicating when a change of owner for objects within database scope occurs.

Rationale:
Use this trace event to determine when the owner of an object inside the database changes. An example usage of this trace event would be when you suspect unauthorized users are changing schema ownership.

Remediation:
Modify your trace to add this event if it contains information you are interested in monitoring. To modify your SQL Server Profiler trace, perform the following steps:

1. Stop your trace if it is currently running. Go to File menu and choose Properties.
2. Choose the Events Selection tab on the Properties screen. Choose the Show All Events dialog box.
3. Choose the SQL Trace events that you wish to capture in your SQL Server profiler Trace.
4. Choose the Show All Columns Dialog box. Add additional data columns as necessary.

Audit:
View the server trace definition to verify that this event is being captured. The trace definition can be viewed from the SQL Server Profiler trace Properties page. To view the trace properties in SQL Server Profiler, go to the File menu and choose Properties. Choose the Events Selection table on the properties screen to view the events and columns currently being captured.

Default Value:
This event is not captured by default.

References:
http://msdn.microsoft.com/en-us/library/ms175481.aspx

1.15.24 *Audit Database Operation (Level 1, Scorable)*

Description:
SQL Server event indicating when operations in a database, such as checkpoint or subscribe query notification have occurred.

Rationale:
Use this trace event to determine when a database property has been altered. An example usage of this trace event would be when you suspect unauthorized users have changed the recovery mode for a database.

Remediation:
Modify your trace to add this event if it contains information you are interested in monitoring. To modify your SQL Server Profiler trace, perform the following steps:

1. Stop your trace if it is currently running. Go to File menu and choose Properties.
2. Choose the Events Selection tab on the Properties screen. Choose the Show All Events dialog box.
3. Choose the SQL Trace events that you wish to capture in your SQL Server profiler Trace.
4. Choose the Show All Columns Dialog box. Add additional data columns as necessary.

Audit:
View the server trace definition to verify that this event is being captured. The trace definition can be viewed from the SQL Server Profiler trace Properties page. To view the trace properties in SQL Server Profiler, go to the File menu and choose Properties. Choose the Events Selection table on the properties screen to view the events and columns currently being captured.

Default Value:
This event is not captured by default.

References:
http://msdn.microsoft.com/en-us/library/ms175481.aspx

1.15.25 *Audit Database Principal Impersonation (Level 1, Scorable)*

Description:
SQL Server event indicating when an impersonation occurs within the database scope, such as `EXECUTE AS <user>` or `SETUSER`.

Rationale:
Use this trace event to determine when a user is impersonated inside the database. An example usage of this trace event would be when you suspect unauthorized users are impersonating other users in a database on your SQL Server instance.

Remediation:
Modify your trace to add this event if it contains information you are interested in monitoring. To modify your SQL Server Profiler trace, perform the following steps:

1. Stop your trace if it is currently running. Go to File menu and choose Properties.
2. Choose the Events Selection tab on the Properties screen. Choose the Show All Events dialog box.
3. Choose the SQL Trace events that you wish to capture in your SQL Server profiler Trace.
4. Choose the Show All Columns Dialog box. Add additional data columns as necessary.

Audit:
View the server trace definition to verify that this event is being captured. The trace definition can be viewed from the SQL Server Profiler trace Properties page. To view the trace properties in SQL Server Profiler, go to the File menu and choose Properties. Choose the Events Selection table on the properties screen to view the events and columns currently being captured.

Default Value:
This event is not captured by default.

References:
http://msdn.microsoft.com/en-us/library/ms175481.aspx

1.15.26 *Audit Database Principal Management (Level 1, Scorable)*

Description:
SQL Server event indicating when you use the `sp_defaultdb` stored procedure, the `sp_defaultlanguage` stored procedure, or the `ALTER LOGIN` statement to modify a property of a login.

Rationale:
Use this trace event to determine when a property of a login has been altered. An example usage of this trace event would be when you suspect unauthorized users are modifying login passwords.

Remediation:
Modify your trace to add this event if it contains information you are interested in monitoring. To modify your SQL Server Profiler trace, perform the following steps:

1. Stop your trace if it is currently running. Go to File menu and choose Properties.
2. Choose the Events Selection tab on the Properties screen. Choose the Show All Events dialog box.
3. Choose the SQL Trace events that you wish to capture in your SQL Server profiler Trace.

4. Choose the Show All Columns Dialog box. Add additional data columns as necessary.

Audit:
View the server trace definition to verify that this event is being captured. The trace definition can be viewed from the SQL Server Profiler trace Properties page. To view the trace properties in SQL Server Profiler, go to the File menu and choose Properties. Choose the Events Selection table on the properties screen to view the events and columns currently being captured.

Default Value:
This event is not captured by default.

References:
http://msdn.microsoft.com/en-us/library/ms175481.aspx

1.15.27 *Audit Database Scope GDR (Level 1, Scorable)*

Description:
SQL Server event indicating that a `GRANT`, `REVOKE`, or `DENY` is issued for a statement permission by any user in Microsoft SQL Server for database-only actions such as granting permissions on a database.

Rationale:
Use this trace event to determine when a `GRANT`, `REVOKE`, or `DENY` is issued for a statement permission by any user in Microsoft SQL Server for database-only actions such as granting permissions on a database.

Remediation:
Modify your trace to add this event if it contains information you are interested in monitoring. To modify your SQL Server Profiler trace, perform the following steps:

1. Stop your trace if it is currently running. Go to File menu and choose Properties.
2. Choose the Events Selection tab on the Properties screen. Choose the Show All Events dialog box.
3. Choose the SQL Trace events that you wish to capture in your SQL Server profiler Trace.
4. Choose the Show All Columns Dialog box. Add additional data columns as necessary.

Audit:
View the server trace definition to verify that this event is being captured. The trace definition can be viewed from the SQL Server Profiler trace Properties page. To view the trace properties in SQL Server Profiler, go to the File menu and choose Properties. Choose

the Events Selection table on the properties screen to view the events and columns currently being captured.

Default Value:
This event is not captured by default.

References:
http://msdn.microsoft.com/en-us/library/ms175481.aspx

1.15.28 *Audit Login Change Password (Level 1, Scorable)*

Description:
SQL Server event indicating when a user changes their Microsoft SQL Server login password.

Rationale:
Use this trace event to determine when a SQL Server password has been changed. An example usage of this trace event would be when you suspect unauthorized users are modifying login passwords.

Remediation:
Modify your trace to add this event if it contains information you are interested in monitoring. To modify your SQL Server Profiler trace, perform the following steps:

1. Stop your trace if it is currently running. Go to File menu and choose Properties.
2. Choose the Events Selection tab on the Properties screen. Choose the Show All Events dialog box.
3. Choose the SQL Trace events that you wish to capture in your SQL Server profiler Trace.
4. Choose the Show All Columns Dialog box. Add additional data columns as necessary.

Audit:
View the server trace definition to verify that this event is being captured. The trace definition can be viewed from the SQL Server Profiler trace Properties page. To view the trace properties in SQL Server Profiler, go to the File menu and choose Properties. Choose the Events Selection table on the properties screen to view the events and columns currently being captured.

Default Value:
This event is not captured by default.

References:
http://msdn.microsoft.com/en-us/library/ms175481.aspx

1.15.29 *Audit Login Change Property (Level 1, Scorable)*

Description:
SQL Server event indicating when a Microsoft Windows login right was added or removed.

Rationale:
Use this trace event to determine when a Microsoft Windows login right was added or removed. An example usage of this trace event would be when you suspect unauthorized users are adding or removing windows logins to your SQL Server instance.

Remediation:
Modify your trace to add this event if it contains information you are interested in monitoring. To modify your SQL Server Profiler trace, perform the following steps:

1. Stop your trace if it is currently running. Go to File menu and choose Properties.
2. Choose the Events Selection tab on the Properties screen. Choose the Show All Events dialog box.
3. Choose the SQL Trace events that you wish to capture in your SQL Server profiler Trace.
4. Choose the Show All Columns Dialog box. Add additional data columns as necessary.

Audit:
View the server trace definition to verify that this event is being captured. The trace definition can be viewed from the SQL Server Profiler trace Properties page. To view the trace properties in SQL Server Profiler, go to the File menu and choose Properties. Choose the Events Selection table on the properties screen to view the events and columns currently being captured.

Default Value:
This event is not captured by default.

References:
http://msdn.microsoft.com/en-us/library/ms175481.aspx

1.15.30 *Audit Login (Level 1, Scorable)*

Description:
SQL Server event indicating when a user has successfully logged in to SQL Server.

Rationale:

Use this trace event to determine when a successful login occurs. An example usage of this trace event would be when you want to modify what users are logging into your database instance.

Remediation:
Modify your trace to add this event if it contains information you are interested in monitoring. To modify your SQL Server Profiler trace, perform the following steps:

1. Stop your trace if it is currently running. Go to File menu and choose Properties.
2. Choose the Events Selection tab on the Properties screen. Choose the Show All Events dialog box.
3. Choose the SQL Trace events that you wish to capture in your SQL Server profiler Trace.
4. Choose the Show All Columns Dialog box. Add additional data columns as necessary.

Audit:
View the server trace definition to verify that this event is being captured. The trace definition can be viewed from the SQL Server Profiler trace Properties page. To view the trace properties in SQL Server Profiler, go to the File menu and choose Properties. Choose the Events Selection table on the properties screen to view the events and columns currently being captured.

Default Value:
This event is not captured by default.

References:
http://msdn.microsoft.com/en-us/library/ms175481.aspx

1.15.31 *Audit Login Failed (Level 1, Scorable)*

Description:
SQL Server event indicating that a user tried to log in to Microsoft SQL Server and failed.

Rationale:
Use this trace event to determine when a login failure occurs. An example usage of this trace event would be when you suspect unauthorized users are attempting brute force login attempts to your SQL Server instance.

Remediation:
Modify your trace to add this event if it contains information you are interested in monitoring. To modify your SQL Server Profiler trace, perform the following steps:

1. Stop your trace if it is currently running. Go to File menu and choose Properties.
2. Choose the Events Selection tab on the Properties screen. Choose the Show All Events dialog box.

3. Choose the SQL Trace events that you wish to capture in your SQL Server profiler Trace.
4. Choose the Show All Columns Dialog box. Add additional data columns as necessary.

Audit:
View the server trace definition to verify that this event is being captured. The trace definition can be viewed from the SQL Server Profiler trace Properties page. To view the trace properties in SQL Server Profiler, go to the File menu and choose Properties. Choose the Events Selection table on the properties screen to view the events and columns currently being captured.

Default Value:
This event is not captured by default.

References:
http://msdn.microsoft.com/en-us/library/ms175481.aspx

1.15.32 *Audit Login GDR Event (Level 1, Scorable)*

Description:
SQL Server event indicating that a Microsoft Windows login right was added or removed.

Rationale:
Use this trace event to determine when a Windows login was added or removed. An example usage of this trace event would be when you suspect unauthorized users are granting, denying, or revoking windows logins on your SQL Server instance.

Remediation:
Modify your trace to add this event if it contains information you are interested in monitoring. To modify your SQL Server Profiler trace, perform the following steps:

1. Stop your trace if it is currently running. Go to File menu and choose Properties.
2. Choose the Events Selection tab on the Properties screen. Choose the Show All Events dialog box.
3. Choose the SQL Trace events that you wish to capture in your SQL Server profiler Trace.
4. Choose the Show All Columns Dialog box. Add additional data columns as necessary.

Audit:
View the server trace definition to verify that this event is being captured. The trace definition can be viewed from the SQL Server Profiler trace Properties page. To view the trace properties in SQL Server Profiler, go to the File menu and choose Properties. Choose the Events Selection table on the properties screen to view the events and columns

currently being captured.

Default Value:
This event is not captured by default.

References:
http://msdn.microsoft.com/en-us/library/ms175481.aspx

1.15.33 *Audit Logout (Level 1, Scorable)*

Description:
SQL Server event indicating that a user has logged out of (logged off) Microsoft SQL Server.

Rationale:
Use this trace event to determine when a server logout occurs.

Remediation:
Modify your trace to add this event if it contains information you are interested in monitoring. To modify your SQL Server Profiler trace, perform the following steps:

1. Stop your trace if it is currently running. Go to File menu and choose Properties.
2. Choose the Events Selection tab on the Properties screen. Choose the Show All Events dialog box.
3. Choose the SQL Trace events that you wish to capture in your SQL Server profiler Trace.
4. Choose the Show All Columns Dialog box. Add additional data columns as necessary.

Audit:
View the server trace definition to verify that this event is being captured. The trace definition can be viewed from the SQL Server Profiler trace Properties page. To view the trace properties in SQL Server Profiler, go to the File menu and choose Properties. Choose the Events Selection table on the properties screen to view the events and columns currently being captured.

Default Value:
This event is not captured by default.

References:
http://msdn.microsoft.com/en-us/library/ms175481.aspx

1.15.34 *Audit Object Derived Permission Event (Level 1, Scorable)*

Description:

SQL Server event indicating when a `CREATE`, `ALTER`, or `DROP` was issued for an object.

Rationale:
Use this trace event to determine when an object is created or altered. This event only occurs if the object does not have permissions or owners directly associated with it.

Remediation:
Modify your trace to add this event if it contains information you are interested in monitoring. To modify your SQL Server Profiler trace, perform the following steps:

1. Stop your trace if it is currently running. Go to File menu and choose Properties.
2. Choose the Events Selection tab on the Properties screen. Choose the Show All Events dialog box.
3. Choose the SQL Trace events that you wish to capture in your SQL Server profiler Trace.
4. Choose the Show All Columns Dialog box. Add additional data columns as necessary.

Audit:
View the server trace definition to verify that this event is being captured. The trace definition can be viewed from the SQL Server Profiler trace Properties page. To view the trace properties in SQL Server Profiler, go to the File menu and choose Properties. Choose the Events Selection table on the properties screen to view the events and columns currently being captured.

Default Value:
This event is not captured by default.

References:
http://msdn.microsoft.com/en-us/library/ms175481.aspx

1.15.35 *Audit Schema Object Access (Level 1, Scorable)*

Description:
SQL Server event indicating when objects permission (such as `SELECT`) is used.

Rationale:
Use this trace event to determine when users are using SELECT statements on tables. An example usage of this trace event would be when you suspect unauthorized users accessing unauthorized tables via SELECT statements.

Remediation:
Modify your trace to add this event if it contains information you are interested in monitoring. To modify your SQL Server Profiler trace, perform the following steps:

1. Stop your trace if it is currently running. Go to File menu and choose Properties.
2. Choose the Events Selection tab on the Properties screen. Choose the Show All Events dialog box.
3. Choose the SQL Trace events that you wish to capture in your SQL Server profiler Trace.
4. Choose the Show All Columns Dialog box. Add additional data columns as necessary.

Audit:
View the server trace definition to verify that this event is being captured. The trace definition can be viewed from the SQL Server Profiler trace Properties page. To view the trace properties in SQL Server Profiler, go to the File menu and choose Properties. Choose the Events Selection table on the properties screen to view the events and columns currently being captured.

Default Value:
This event is not captured by default.

References:
http://msdn.microsoft.com/en-us/library/ms175481.aspx

1.15.36 *Audit Schema Object GDR (Level 1, Scorable)*

Description:
SQL Server event indicating whenever a `GRANT`, `DENY`, or `REVOKE` statement is issued for schema object permission by any user in Microsoft SQL Server.

Rationale:
Use this trace event to determine when `GRANT`, `DENY`, or `REVOKE` permissions are given. An example usage of this trace event would be when you suspect unauthorized users are granting, denying, or revoking permissions on a schema object.

Remediation:
Modify your trace to add this event if it contains information you are interested in monitoring. To modify your SQL Server Profiler trace, perform the following steps:

1. Stop your trace if it is currently running. Go to File menu and choose Properties.
2. Choose the Events Selection tab on the Properties screen. Choose the Show All Events dialog box.
3. Choose the SQL Trace events that you wish to capture in your SQL Server profiler Trace.
4. Choose the Show All Columns Dialog box. Add additional data columns as necessary.

Audit:
View the server trace definition to verify that this event is being captured. The trace definition can be viewed from the SQL Server Profiler trace Properties page. To view the trace properties in SQL Server Profiler, go to the File menu and choose Properties. Choose the Events Selection table on the properties screen to view the events and columns currently being captured.

Default Value:
This event is not captured by default.

References:
http://msdn.microsoft.com/en-us/library/ms175481.aspx

1.15.37 *Audit Schema Object Management (Level 1, Scorable)*

Description:
SQL Server event indicating when server objects are created, altered, or dropped.

Rationale:
Use this trace event to determine when server objects are created, altered, or dropped. An example usage of this trace event would be when you suspect unauthorized users are dropping server objects, such as linked servers.

Remediation:
Modify your trace to add this event if it contains information you are interested in monitoring. To modify your SQL Server Profiler trace, perform the following steps:

1. Stop your trace if it is currently running. Go to File menu and choose Properties.
2. Choose the Events Selection tab on the Properties screen. Choose the Show All Events dialog box.
3. Choose the SQL Trace events that you wish to capture in your SQL Server profiler Trace.
4. Choose the Show All Columns Dialog box. Add additional data columns as necessary.

Audit:
View the server trace definition to verify that this event is being captured.

Default Value:
This event is not captured by default.

References:
http://msdn.microsoft.com/en-us/library/ms175481.aspx

1.15.38 *Audit Schema Object Take Ownership (Level 1, Scorable)*

Description:
SQL Server event indicating when the permissions to change the owner of schema object (such as a table, procedure, or function) is checked. This happens when the `ALTER AUTHORIZATION` statement is used to assign an owner to an object.

Rationale:
Use this trace event to determine when permissions are given to change the owner of an object. An example usage of this trace event would be when you suspect unauthorized users are changing table object owners. `ALTER AUTHORIZATION` statements invoke this event.

Remediation:
Modify your trace to add this event if it contains information you are interested in monitoring. To modify your SQL Server Profiler trace, perform the following steps:

1. Stop your trace if it is currently running. Go to File menu and choose Properties.
2. Choose the Events Selection tab on the Properties screen. Choose the Show All Events dialog box.
3. Choose the SQL Trace events that you wish to capture in your SQL Server profiler Trace.
4. Choose the Show All Columns Dialog box. Add additional data columns as necessary.

Audit:
View the server trace definition to verify that this event is being captured. The trace definition can be viewed from the SQL Server Profiler trace Properties page. To view the trace properties in SQL Server Profiler, go to the File menu and choose Properties. Choose the Events Selection table on the properties screen to view the events and columns currently being captured.

Default Value:
This event is not captured by default.

References:
http://msdn.microsoft.com/en-us/library/ms175481.aspx

1.15.39 *Audit Server Alter Trace (Level 1, Scorable)*

Description:
SQL Server event indicating when `ALTER TRACE` permissions have been checked. Statements that check for `ALTER TRACE` include those used to create or configure a trace, or to set a filter on a trace.

Rationale:
Use this trace event to determine when a server trace is created or altered. An example usage of this trace event would be when you suspect unauthorized users are granting permissions to run SQL Server traces.

Remediation:
Modify your trace to add this event if it contains information you are interested in monitoring. To modify your SQL Server Profiler trace, perform the following steps:

1. Stop your trace if it is currently running. Go to File menu and choose Properties.
2. Choose the Events Selection tab on the Properties screen. Choose the Show All Events dialog box.
3. Choose the SQL Trace events that you wish to capture in your SQL Server profiler Trace.
4. Choose the Show All Columns Dialog box. Add additional data columns as necessary.

Audit:
View the server trace definition to verify that this event is being captured. The trace definition can be viewed from the SQL Server Profiler trace Properties page. To view the trace properties in SQL Server Profiler, go to the File menu and choose Properties. Choose the Events Selection table on the properties screen to view the events and columns currently being captured.

Default Value:
This event is not captured by default.

References:
http://msdn.microsoft.com/en-us/library/ms175481.aspx

1.15.40 *Audit Server Object GDR (Level 1, Scorable)*

Description:
SQL Server event indicating whenever a `GRANT`, `DENY` or `REVOKE`, is issued for a server object permission by any user in Microsoft SQL Server.

Rationale:
Use this trace event to determine when a `GRANT`, `DENY`, or `REVOKE` statement is issued on a server object. An example usage of this trace event would be when you suspect unauthorized users are granting, denying, permissions on server objects, such as linked servers.

Remediation:

Modify your trace to add this event if it contains information you are interested in monitoring. To modify your SQL Server Profiler trace, perform the following steps:

1. Stop your trace if it is currently running. Go to File menu and choose Properties.
2. Choose the Events Selection tab on the Properties screen. Choose the Show All Events dialog box.
3. Choose the SQL Trace events that you wish to capture in your SQL Server profiler Trace.
4. Choose the Show All Columns Dialog box. Add additional data columns as necessary.

Audit:
View the server trace definition to verify that this event is being captured. The trace definition can be viewed from the SQL Server Profiler trace Properties page. To view the trace properties in SQL Server Profiler, go to the File menu and choose Properties. Choose the Events Selection table on the properties screen to view the events and columns currently being captured.

Default Value:
This event is not captured by default.

References:
http://msdn.microsoft.com/en-us/library/ms175481.aspx

1.15.41 *Audit Server Object Management (Level 1, Scorable)*

Description:
SQL Server event indicating when a `CREATE`, `ALTER`, or `DROP` statement occurs for server objects.

Rationale:
Use this trace event to determine when a server object is affected. An example usage of this trace event would be when you suspect unauthorized users are creating, altering or dropping server objects, such as linked servers.

Remediation:
Modify your trace to add this event if it contains information you are interested in monitoring. To modify your SQL Server Profiler trace, perform the following steps:

1. Stop your trace if it is currently running. Go to File menu and choose Properties.
2. Choose the Events Selection tab on the Properties screen. Choose the Show All Events dialog box.
3. Choose the SQL Trace events that you wish to capture in your SQL Server profiler Trace.

4. Choose the Show All Columns Dialog box. Add additional data columns as necessary.

Audit:
View the server trace definition to verify that this event is being captured. The trace definition can be viewed from the SQL Server Profiler trace Properties page. To view the trace properties in SQL Server Profiler, go to the File menu and choose Properties. Choose the Events Selection table on the properties screen to view the events and columns currently being captured.

Default Value:
This event is not captured by default.

References:
http://msdn.microsoft.com/en-us/library/ms175481.aspx

1.15.42 *Audit Server Object Take Ownership (Level 1, Scorable)*

Description:
SQL Server event indicating when the owner is changed for objects in server scope.

Rationale:
Use this trace event to determine when a server object's owner is changed. An example usage of this trace event would be when you suspect unauthorized users alter ownership permissions on a server object, such as an Endpoint.

Remediation:
Modify your trace to add this event if it contains information you are interested in monitoring. To modify your SQL Server Profiler trace, perform the following steps:

1. Stop your trace if it is currently running. Go to File menu and choose Properties.
2. Choose the Events Selection tab on the Properties screen. Choose the Show All Events dialog box.
3. Choose the SQL Trace events that you wish to capture in your SQL Server profiler Trace.
4. Choose the Show All Columns Dialog box. Add additional data columns as necessary.

Audit:
View the server trace definition to verify that this event is being captured. The trace definition can be viewed from the SQL Server Profiler trace Properties page. To view the trace properties in SQL Server Profiler, go to the File menu and choose Properties. Choose the Events Selection table on the properties screen to view the events and columns currently being captured.

Default Value:
This event is not captured by default.

References:
http://msdn.microsoft.com/en-us/library/ms175481.aspx

1.15.43 *Audit Server Operation (Level 1, Scorable)*

Description:
SQL Server event indicating when Security Audit operations such as altering settings, resources, external access, or authorization are used.

Rationale:
Use this trace event to determine when server settings are altered. An example usage of this trace event would be when you suspect unauthorized users are performing security audit operations, such as allowing or prohibiting external access.

Remediation:
Modify your trace to add this event if it contains information you are interested in monitoring. To modify your SQL Server Profiler trace, perform the following steps:

1. Stop your trace if it is currently running. Go to File menu and choose Properties.
2. Choose the Events Selection tab on the Properties screen. Choose the Show All Events dialog box.
3. Choose the SQL Trace events that you wish to capture in your SQL Server profiler Trace.
4. Choose the Show All Columns Dialog box. Add additional data columns as necessary.

Audit:
View the server trace definition to verify that this event is being captured. The trace definition can be viewed from the SQL Server Profiler trace Properties page. To view the trace properties in SQL Server Profiler, go to the File menu and choose Properties. Choose the Events Selection table on the properties screen to view the events and columns currently being captured.

Default Value:
This event is not captured by default.

References:
http://msdn.microsoft.com/en-us/library/ms175481.aspx

1.15.44 *Audit Server Principal Impersonation (Level 1, Scorable)*

Description:
SQL Server event indicating when impersonation occurs within server scope, such as `EXECUTE AS <login>`.

Rationale:
Use this trace event to determine when login impersonation occurs. An example usage of this trace event would be when you suspect unauthorized users are making user of Impersonation through the Execute As command.

Remediation:
Modify your trace to add this event if it contains information you are interested in monitoring. To modify your SQL Server Profiler trace, perform the following steps:

1. Stop your trace if it is currently running. Go to File menu and choose Properties.
2. Choose the Events Selection tab on the Properties screen. Choose the Show All Events dialog box.
3. Choose the SQL Trace events that you wish to capture in your SQL Server profiler Trace.
4. Choose the Show All Columns Dialog box. Add additional data columns as necessary.

Audit:
View the server trace definition to verify that this event is being captured. The trace definition can be viewed from the SQL Server Profiler trace Properties page. To view the trace properties in SQL Server Profiler, go to the File menu and choose Properties. Choose the Events Selection table on the properties screen to view the events and columns currently being captured.

Default Value:
This event is not captured by default.

References:
http://msdn.microsoft.com/en-us/library/ms175481.aspx

1.15.45 *Audit Server Principal Management (Level 1, Scorable)*

Description:
SQL Server event indicating when server principals are created, altered, or dropped.

Rationale:
Use this trace event to determine when a server principal (login) is created, altered, or dropped. An example usage of this trace event would be when you suspect unauthorized users are creating, altering, or dropping server principals (logins).

Remediation:
Modify your trace to add this event if it contains information you are interested in monitoring. To modify your SQL Server Profiler trace, perform the following steps:

1. Stop your trace if it is currently running. Go to File menu and choose Properties.
2. Choose the Events Selection tab on the Properties screen. Choose the Show All Events dialog box.
3. Choose the SQL Trace events that you wish to capture in your SQL Server profiler Trace.
4. Choose the Show All Columns Dialog box. Add additional data columns as necessary.

Audit:
View the server trace definition to verify that this event is being captured. The trace definition can be viewed from the SQL Server Profiler trace Properties page. To view the trace properties in SQL Server Profiler, go to the File menu and choose Properties. Choose the Events Selection table on the properties screen to view the events and columns currently being captured.

Default Value:
This event is not captured by default.

References:
http://msdn.microsoft.com/en-us/library/ms175481.aspx

1.15.46 *Audit Server Scope GDR (Level 1, Scorable)*

Description:
SQL Server event indicating when a `GRANT`, `REVOKE`, or `DENY` is issued for permissions in the server scope, such as creating a login.

Rationale:
Use this trace event to determine when a `GRANT`, `DENY`, OR `REVOKE` event occurs at the server level. An example usage of this trace event would be when you suspect unauthorized users are granting, denying, or revoking login permissions.

Remediation:
Modify your trace to add this event if it contains information you are interested in monitoring. To modify your SQL Server Profiler trace, perform the following steps:

1. Stop your trace if it is currently running. Go to File menu and choose Properties.
2. Choose the Events Selection tab on the Properties screen. Choose the Show All Events dialog box.

3. Choose the SQL Trace events that you wish to capture in your SQL Server profiler Trace.
4. Choose the Show All Columns Dialog box. Add additional data columns as necessary.

Audit:
View the server trace definition to verify that this event is being captured. The trace definition can be viewed from the SQL Server Profiler trace Properties page. To view the trace properties in SQL Server Profiler, go to the File menu and choose Properties. Choose the Events Selection table on the properties screen to view the events and columns currently being captured.

Default Value:
This event is not captured by default.

References:
http://msdn.microsoft.com/en-us/library/ms175481.aspx

1.15.47 *Audit Server Starts and Stops (Level 1, Scorable)*

Description:
SQL Server event indicating when the Microsoft SQL Server service state is modified.

Rationale:
Use this trace event to determine when the SQL Server service state has been modified.

Remediation:
Modify your trace to add this event if it contains information you are interested in monitoring. To modify your SQL Server Profiler trace, perform the following steps:

1. Stop your trace if it is currently running. Go to File menu and choose Properties.
2. Choose the Events Selection tab on the Properties screen. Choose the Show All Events dialog box.
3. Choose the SQL Trace events that you wish to capture in your SQL Server profiler Trace.
4. Choose the Show All Columns Dialog box. Add additional data columns as necessary.

Audit:
View the server trace definition to verify that this event is being captured. The trace definition can be viewed from the SQL Server Profiler trace Properties page. To view the trace properties in SQL Server Profiler, go to the File menu and choose Properties. Choose the Events Selection table on the properties screen to view the events and columns currently being captured.

Default Value:
This event is not captured by default.

References:
http://msdn.microsoft.com/en-us/library/ms175481.aspx

1.15.48 *Audit Statement Permission Event (Level 1, Scorable)*

Description:
SQL Server event indicating when statement permission has been used.

Rationale:
Use this trace event to determine when a statement permission has been issued. An example usage of this trace event would be when you suspect unauthorized users are granting CREATE TABLE permissions.

Remediation:
Modify your trace to add this event if it contains information you are interested in monitoring. To modify your SQL Server Profiler trace, perform the following steps:

1. Stop your trace if it is currently running. Go to File menu and choose Properties.
2. Choose the Events Selection tab on the Properties screen. Choose the Show All Events dialog box.
3. Choose the SQL Trace events that you wish to capture in your SQL Server profiler Trace.
4. Choose the Show All Columns Dialog box. Add additional data columns as necessary.

Audit:
View the server trace definition to verify that this event is being captured. The trace definition can be viewed from the SQL Server Profiler trace Properties page. To view the trace properties in SQL Server Profiler, go to the File menu and choose Properties. Choose the Events Selection table on the properties screen to view the events and columns currently being captured.

Default Value:
This event is not captured by default.

References:
http://msdn.microsoft.com/en-us/library/ms175481.aspx

1.16 Backup and Disaster Recovery Procedures

1.16.1 *Backups – General (Level 1, Not Scorable)*

Description:
Use Full database backups combined with differential or transaction log backups to restore the database to a specific point in time.

Rationale:

Database backups should be made to another server or disk that is not physically attached to the same server as the database. This will reduce the risk of total loss in case of disk failure.

Remediation:
If a database recovery mode is not set to Full Recovery, it cannot be restored to a point in time. If a point in time recovery is not necessary make sure your recovery mode is set to Simple. If the ability to recover to a point in time is necessary then ensure that your database is set to the full recovery mode.

Use the following code to set the recovery mode for a database to full.

```
ALTER DATABASE <database_name>
SET RECOVERY FULL;
```

Audit:
Use the following TSQL snippet to determine the recovery mode for all databases on the instance.

```
SELECT name, recovery_model_desc
FROM sys.databases;
```

Default Value:
SQL Server user databases are set to the Full Recovery mode by default.

References:
http://msdn.microsoft.com/en-us/library/ms190982.aspx

1.16.2 *System databases (Level 1, Not Scorable)*

Description:
It is important to include the system databases in your backup plan i.e. the `master`, `msdb` and `model` databases.

Rationale:
The `master` database contains login and system settings and is frequent backup is crucial. The `msdb` database contains SQL Agent jobs along with backup and restore information and is necessary to backup. The `model` database is a template database which is used to create

new databases. Backup is really only necessary when changes are made. The `tempdb` does not contain permanent data nor is it not possible to back it up.

Remediation:
Add the master and `msdb` databases to your regular backup routine. Include the `model` database if changes are frequently made.

Audit:
Use the following code snippet to identify the last backup dates for the system databases.

```
SELECT
      database_name,
      lastbackuptime = MAX(backup_finish_date)
FROM msdb.dbo.backupset
WHERE database_name IN('master', 'msdb', 'model')
GROUP BY database_name;
```

Default Value:
User and system databases are not backed up by default.

References:
http://msdn.microsoft.com/en-us/library/ms178028.aspx

1.16.3 *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 `master` database in your database maintenance plans, being doing 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:
http://articles.techrepublic.com.com/5100-10878_11-6105067.html

1.16.4 *Backing up MSDB database (Level 1, Not Scorable)*

Description:
Backup the `msdb` database when an alerts, job, job schedules or operators have been created, modified, or removed.

Rationale:
The `msdb` database holds information regarding SQL Server agent along with backup and restore information. Regular backups of this database should be performed to ensure this information is not lost.

Remediation:
If you are not currently including the backup model database in your database maintenance plans, being doing so immediately. Typically this database does not need to be backed up more than once or twice per week.

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

1.16.5 *Backup Media (Level 2, Not Scorable)*

Description:
Add a password to protect the backup media.

Rationale:
The protection provided by this password is weak. It is intended to prevent an incorrect restore using SQL Server tools by authorized or unauthorized users. It does not prevent the reading of the backup data by another method or the replacement of the password. This feature will be removed in the next version of Microsoft SQL Server. Avoid using this feature in new development work, and plan to modify applications that currently use this feature.

Remediation:
Consider placing passwords on backup files so that an incorrect restore does not occur.

Audit:
You can use the `RESTORE FILELISTONLY` command to determine if a particular backup file is password protected. If the file is protected, an error will occur:

```
RESTORE FILELISTONLY FROM DISK = 'C:\DatabaseName.bak';
```

Default Value:
SQL Server backup files are not password protected by default.

References:
http://msdn.microsoft.com/en-us/library/ms190964.aspx

1.16.6 *Restrict access to backup files to System Administrators (Level 1, Scorable)*

Description:
Restrict access to the backup files to System Administrators.

Rationale:
Backup files contain data that is owned by the company. These backups may contain very sensitive information such as credit card numbers or social security numbers. It is essential that backup files are not able to be copied by unauthorized personnel. In addition, backups are required to restore a database in the event of an emergency. Unauthorized personnel must not be allowed to modify, move, or delete these files.

Remediation:
Remove any all permissions for users who are not system administrators.

Audit:
Review the security for the backup destination folders. Review user groups and members that have access to these folders.

1.16.7 *Restrict restore permissions to Database Administrators (Level 1, Not Scorable)*

Description:
Restrict restore permissions to Database Administrators.

Rationale:
RESTORE permissions default to members of the `sysadmin` and `dbcreator` fixed server roles, and the owner (`dbo`) of the database. Holders of the `CREATE DATABASE` permission can restore a database that does not currently exist.

Remediation:

Restrict restore permissions to only those individuals who need to restore databases. This permission allows a user to overwrite an existing database, which can cause destruction in a production environment.

Audit:
Run the following code snippet to identity any users or groups who are members of the `db_owner` database role or `dbcreator` server role.

```
SELECT DBCreator = mem.name
from sys.server_role_members sm
JOIN sys.server_principals mem ON sm.member_principal_id = mem.principal_id
JOIN sys.server_principals rol ON sm.role_principal_id = rol.principal_id
WHERE rol.name = 'dbcreator'

SELECT DBOwner = mem.name
from sys.database_role_members sm
JOIN sys.database_principals mem ON sm.member_principal_id = mem.principal_id
JOIN sys.database_principals rol ON sm.role_principal_id = rol.principal_id
WHERE rol.name = 'db_owner'
```

Default Value:
By default, only the `sysadmin` account, `db_owner` database role, and `dbcreator` server role are able to restore databases.

1.16.8 *Run the Microsoft Baseline Security Analyzer (Level 1, Not Scorable)*

Description:
Run the Microsoft Baseline Security Analyzer weekly and follow the security recommendations as closely as possible to secure the operating system.

Rationale:
Microsoft Baseline Security Analyzer (MBSA) is an easy-to-use tool that helps small and medium businesses determine their security state in accordance with Microsoft security recommendations and offers specific remediation guidance.

Remediation:
Download and install the MBSA. Run regular audits to determine if your system is up to the recommendations of the MBSA.

Audit:
Download the tool from the References link below to perform a system security analysis.

Default Value:
Not applicable.

References:
http://technet.microsoft.com/en-us/security/cc184923.aspx

1.16.9 *Run the SQL Best Practices Analyzer (Level 1, Not Scorable)*

Description:
Run the SQL Best Practices Analyzer regularly and note any changes to the environment.

Remediation:
The SQL Server 2005 Best Practices Analyzer (BPA) gathers data from Microsoft Windows and SQL Server configuration settings. BPA uses a predefined list of SQL Server 2005 recommendations and best practices to determine if there are potential issues in the database environment. Run this tool regularly to identity any potential environment security issues.

Audit:
Not applicable.

Default Value:
This application is not installed by default.

References:
http://www.microsoft.com/downloads/details.aspx?FamilyId=DA0531E4-E94C-4991-82FA-F0E3FBD05E63&displaylang=en

1.16.10 *Enable Password Policy Enforcement (Level 1, Scorable)*

Description:
When SQL Server is running on a Windows Server 2003 or later machine, Windows password policy mechanisms may be used to ensure that SQL Server passwords are complex. Enforce Windows password policies for SQL Server user accounts.

Rationale:
Enforcing the security policies for SQL Server passwords ensures that SQL Server account passwords are not easily cracked.

Remediation:
If the SQL Server instance is on a Windows Server 2003 or later machine, ensure that all SQL Login security is enforced through Windows policies.

Audit:
Run the following code snippet to determine the policy enforcement status of the SQL Server logins in the instance.

```
SELECT SQLLoginName = name, PolicyEnforced = is_policy_checked
FROM sys.sql_logins;
```

Default Value:

Enforce password policies are on my default.

References:
http://msdn.microsoft.com/en-us/library/ms189751(SQL.90).aspx

1.16.11 *Periodic scan of Role Members (Level 1, Not Scorable)*

Description:
Periodically scan fixed server and database roles to ensure that only trusted individuals are members.

Rationale:
This action should be performed to ensure that those with the necessary privileges have not modified security settings that you are not aware of.

Remediation:
Remove permissions for users as necessary.

Audit:
Run the following snippet to determine all role members for a given database:

```
SELECT RoleName = p.name, MemberName = m.name
FROM sys.database_role_members rm
JOIN sys.database_principals p ON rm.role_principal_id = p.principal_id
JOIN sys.database_principals m ON rm.member_principal_id = m.principal_id
ORDER BY p.name;
```

Default Value:
Not applicable.

References:
http://msdn.microsoft.com/en-us/library/ms189780.aspx

1.16.12 *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. 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.aspx

1.17 Replication

1.17.1 *SQL Server Agent service account (Level 1, Scorable)*

Description:
Configure replication agents to use a Windows account rather than a SQL Server Agent account. Grant only the required permissions to each agent.

Rationale:
Use Windows Authentication for all replication agent connections.

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.

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 are created.

References:
http://msdn.microsoft.com/en-us/library/ms152501.aspx

1.17.2 *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.

Rationale:
The permissions needed to support and administer replication are assigned to `sysadmin`, `db_owner` and `replmonitor` by default.

Remediation:
Review the individuals responsible for administering replication. Ensure these individuals are members of the necessary groups to perform their duties.

Audit:
Review the system permissions of those individuals 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.aspx

1.17.3 *Snapshot share folder (Level 1, Scorable)*

Description:
Store the snapshot folder, which houses a snapshot of the replicated changes, on an explicit share and not an administrative share.

Remediation:
Storing snapshot folders on a network share allows you to easily assign explicit permissions to the SQL Server agent to access.

Audit:
Review permissions on the network share where snapshot files are created.

Default Value:
Not applicable.

References:
http://msdn.microsoft.com/en-us/library/ms151151.aspx

1.17.4 *Publication Access List (Level 1, Scorable)*

Description:
The domain accounts used by the SQL Server Agent service and the Replication proxy account must be entered in the Publication Access List so that all replication agents will be able to participate in the replication process.

Remediation:

Add any necessary account that is not currently present to the Publication Access list.

Audit:
The publication access list may be viewed under the properties of the publication in SQL Server Management Studio. To view the publication properties, 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 publisher.
3. Navigate to the Replication folder and expand the Local Publications dropdown.
4. Right click the publication and select Properties.
5. The members of the PAC may be viewed under the Publication Access List tab.

Default Value:
By default, the necessary accounts are added to the publication access list when using the SQL Server replication wizard to configure a publication.

References:
http://technet.microsoft.com/en-us/library/ms152764(SQL.90).aspx

1.17.5 *Secure Communications (Level 1, Not Scorable)*

Description:
Use secure connections, such as VPN or proxy servers, for all replication over the Internet.

Rationale:
Using a secure communication connection will ensure that replicated data is not compromised 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 a secure channel is currently available to secure replicated data.

Default Value:
By default, replication does not require a secure communication channel to transmit data.

References:
http://msdn.microsoft.com/en-us/library/ms151172.aspx

1.17.6 *Database connections (Level 1, Scorable)*

Description:
Configure the database connections for replication agents to use Windows authenticated logons.

Rationale:
Run each replication agent under a different Windows account, and use Windows Authentication for all replication agent connections. Windows authentication is a more secure form of authentication than SQL Server authentication.

Remediation:
Review the SQL Agent accounts and consider using Windows accounts in those instances where the agent currently uses a SQL Server account.

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 are created.

References:
http://msdn.microsoft.com/en-us/library/ms152501.aspx

1.17.7 *Filtering (Level 1, Scorable)*

Description:
Employ replication filters to protect the data where applicable.

Rationale:
Use replication filters to bypass replicating sensitive data, such as a passwords, social security numbers, etc.

Remediation:
Evaluate the data that you are currently replicating or intend to replicate. Consider using filters so that sensitive fields are not replicated.

Audit:
Review your current replication topology to identity those tables that columns that contain sensitive information.

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/ms151198.aspx

1.17.8 *Distribution databases (Level 1, Not Scorable)*

Description:
All distribution databases and snapshot files must be located in protected and audited locations.

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. Review the snapshot folder security scheme.

Default Value:
Not applicable.

References:
http://msdn.microsoft.com/en-us/library/ms151227.aspx

1.18 Application Development Best Practices

1.18.1 *Ownership Chaining (Level 1, Not Scorable)*

Description:
Avoid using cross database ownership chaining on the SQL Server instance.

Rationale:
SQL Server cross database ownership chaining is a server level option that allows for the bypassing of permissions between databases if the object is owned by the same login account. Using database ownership chaining allows for the highly-privileged actions that

logins can perform as a result of enabling the option.

Remediation:
Disable cross database ownership chaining if it is enabled and not needed. To disable this option, execute the following code snippet:

```
EXECUTE sp_configure 'cross db ownership chaining',0
RECONFIGURE WITH OVERRIDE
```

Audit:
To determine if cross database ownership chaining is enabled for the SQL Server instance, execute the following code snippet:

```
EXECUTE sp_configure 'cross db ownership chaining';
```

A run_value of 1 indicates that the option is enabled.

Default Value:
Cross database ownership chaining is not enabled by default.

References:
http://msdn.microsoft.com/en-us/library/ms188676.aspx

1.18.2 *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:

Specific permissions are not assigned to database users by default.

References:
http://msdn.microsoft.com/en-us/library/aa905188%28SQL.80%29.aspx

1.18.3 *Encrypted connections (Level 1, Not Scorable)*

Description:
Enable encrypted connections between the user and the server.

Rationale:
Consider allowing only encrypted connections. When allowing SQL Server authentication, encrypt either the network layer with IPSec or the session with SSL.

Remediation:
Consider your network needs. If secure communications are a requirement from the SQL Server machine then consider encrypting SQL Server connections.

Audit:
Perform the following steps to determine if protocol encryption is currently being used:

1. Run `compmgmt.msc` from the Run menu.
2. Expand Services and Applications.
3. Expand SQL Server Network Configuration.
4. Right-click Protocols for SQL Server and select Properties.
5. The Force Encryption option determines if Encryption is being used.

Default Value:
By default, connections to SQL Server are not encrypted.

References:
http://support.microsoft.com/kb/316898
http://msdn.microsoft.com/en-us/library/ms189067.aspx

1.18.4 *Error Handling (Level 1, Not Scorable)*

Description:
Do not propagate errors back to the user.

Rationale:
Log errors or transmit them to the system administrator. Displaying system error messages to the user can give details of the underlying database system.

Remediation:
Ensure that the web server only displays 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 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:
http://technet.microsoft.com/en-us/library/cc778973(WS.10).aspx

1.18.5 *User Input (Level 1, Not Scorable)*

Description:
Prevent SQL injection by validating all user input before transmitting it to the server.

Rationale:
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.

Remediation:
Use stored procedures for all database interaction. Using stored procedures parameterizes user input, drastically minimizing any risk for SQL injection.

Audit:
Check with the application teams to ensure any database interaction is through the use of stored procedures and not dynamic SQL. Revoke any `INSERT`, `UPDATE`, or `DELETE` privileges to users so that modifications to data must be done through stored procedures.

Default Value:
Not applicable.

References:
http://msdn.microsoft.com/en-us/library/a0z2h4sw(VS.71).aspx

1.18.6 *Developer awareness (Level 1, Not Scorable)*

Description:
Increase awareness of issues such as cross-site scripting, buffer overflows, SQL injection and dangerous APIs.

Rationale:
The more developers are aware of potential threats to the database through coding flaws, the less likely the events are to happen.

Remediation:

Have developers perform code reviews of each other's code along with you performing code reviews on their database code. Encourage the developers and the company to invest in training opportunities.

Audit:
Access the general knowledge of the application development team to get an understanding of their database security consciousness.

Default Value:
Not applicable.

References:
http://blogs.msdn.com/sqlsecurity/

1.18.7 *Developer awareness (Level 1, Not Scorable)*

Description:
Identify categories of threats that apply to your application, such as denial of service, escalation of privileges, spoofing, data tampering, information disclosure and repudiation.

Rationale:
The more developers are aware of potential threats to the database through coding flaws, the less likely the events are to happen.

Remediation:
Have developers perform code reviews of each other's code along with you performing code reviews on their database code. Encourage the developers and the company to invest in training opportunities.

Audit:
Access the general knowledge of the application development team to get an understanding of their database security consciousness.

Default Value:
Not applicable.

References:
http://blogs.msdn.com/sqlsecurity/

1.18.8 *Security reviews (Level 1, Not Scorable)*

Description:
Add security reviews to all stages of the application development lifecycle (from design to testing).

Rationale:

Performing security reviews from product inception to completion will ensure that security threats are minimized.

Remediation:
Assemble a group of developers from each development team to server as a security team. This team is responsible for code, network, and database security for the entire application. This team managers and performs security reviews with all development teams.

Audit:
The security team should perform period audits of system security to ensure thoroughness.

Default Value:
Not applicable.

References:
http://msdn.microsoft.com/en-us/library/ms998364.aspx

1.18.9 *Distributing SQLEXPRESS (Level 1, Not Scorable)*

Description:
If you distribute SQLEXPRESS, install SQLEXPRESS using Windows security mode as the default. Never install a blank `sa` password. Use the Microsoft Installer to install SQLEXPRESS.

Rationale:
Installing SQL Express in mixed authentication mode with a weak `sa` password leaves the data and potentially the network open to attack. Windows authentication significantly minimizes this risk.

Remediation:
If you are actively using SQL Server authentication in your SQL Express environment, ensure that the `sa` password is very strong. If you are not using SQL Authentication, switch the authentication mode to Windows only.

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:
The default SQL Server authentication mode is Windows authentication only.

References:
http://msdn.microsoft.com/en-us/library/ms165639(SQL.90).aspx

1.18.10 *Net-Libraries (Level 1, Not Scorable)*

Description:
If SQL Server Express Edition will operate as a local data store, disable any unnecessary client protocols.

Rationale:
If SQL Server will not require outside connections, ensure that any unnecessary client protocols are disabled.

Remediation:
If client applications are only ran from the PC where SQL Express is installed, ensure that only the Shared Memory client protocol is enabled.

Audit:
To determine the settings of the client protocols, perform the following steps:

1. Open Computer Manager using the following Run command: `compmgmt.msc`.
2. Expand Services and Applications.
3. Expand SQL Server Configuration Manager.
4. Expand SQL Server Network Configuration and select Protocols for SQL Server.
5. The protocols will be listed on the right pane of the screen. They may be enabled or disabled there.

Default Value:
Remote connections to the database engine are disabled by default in all editions of SQL Server 2005 and later.

References:
http://msdn.microsoft.com/en-us/library/ms187662.aspx
http://msdn.microsoft.com/en-us/library/ms187892.aspx

1.18.11 *Customer awareness (Level 1, Not Scorable)*

Description:
Let your customers know 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. SQL Express edition must be configured in your application so that it is distributed.

References:
http://msdn.microsoft.com/en-us/library/ms165639(SQL.90).aspx

1.19 Surface Area Configuration Tool

1.19.1 *Ad Hoc Remote Queries (Level 1, Scorable)*

Description:
Disable Ad Hoc Remote Queries where not required

Rationale:
Ad hoc distributed queries use the `OPENROWSET` and `OPENDATASOURCE` functions to connect to remote data sources using OLE DB. Use this dialog to configure support for ad hoc remote queries.

Remediation:
View the status of this feature in the Surface Area Configuration tool. Disable this feature if it is enabled and not required.

Audit:
To view the status of this option, perform the following steps:

1. Open the SQL Server Surface Area Configuration tool. This can be run from the SQL Server folder in the Start Menu or you can run the `SqlSAC.exe` tool from the SQL Server Program files installation path, such as: `C:\Program Files\Microsoft SQL Server\90\Shared\SqlSAC.exe`
2. Select the "Surface Area Configuration for Features" option under the Configure Surface Area options list.
3. Select the "Ad Hoc Remote Queries" option under the Database Engine dropdown list.

Default Value:
This feature is disabled by default in new installations of SQL Server.

References:
http://msdn.microsoft.com/en-us/library/ms189978(SQL.90).aspx

1.19.2 *CLR Integration (Level 1, Scorable)*

Description:
Disable CLR Integration where not required

Rationale:
Execution of user assemblies inside Microsoft SQL Server 2005 is disabled by default. Use this dialog to configure support for the common language runtime (CLR).

Remediation:
View the status of this feature in the Surface Area Configuration tool. Disable this feature if it is enabled and not required.

Audit:
To view the status of this option, perform the following steps:

1. Open the SQL Server Surface Area Configuration tool. This can be run from the SQL Server folder in the Start Menu or you can run the `SqlSAC.exe` tool from the SQL Server Program files installation path, such as: `C:\Program Files\Microsoft SQL Server\90\Shared\SqlSAC.exe`
2. Select the "Surface Area Configuration for Features" option under the Configure Surface Area options list.
3. Select the "CLR Integration" option under the Database Engine dropdown list.

Default Value:
This feature is disabled by default in new installations of SQL Server.

References:
http://msdn.microsoft.com/en-us/library/ms180219(SQL.90).aspx

1.19.3 *DAC (Level 1, Scorable)*

Description:
Disable the remote Dedicated Administrator Connection where not required. Enabling remote DAC is required for clustered installations.

Rationale:
The dedicated administrator connection (DAC) allows an administrator to connect to a server to execute diagnostic functions or Transact-SQL statements. The DAC can be used when the Database Engine will not respond to regular connections.

Remediation:

View the status of this feature in the Surface Area Configuration tool. Disable this feature if it is enabled and not required.

Audit:
To view the status of this option, perform the following steps:

1. Open the SQL Server Surface Area Configuration tool. This can be run from the SQL Server folder in the Start Menu or you can run the `SqlSAC.exe` tool from the SQL Server Program files installation path, such as: `C:\Program Files\Microsoft SQL Server\90\Shared\SqlSAC.exe`
2. Select the "Surface Area Configuration for Features" option under the Configure Surface Area options list.
3. Select the "DAC" option under the Database Engine dropdown list.

Default Value:
This feature is disabled by default in new installations of SQL Server.

References:
http://msdn.microsoft.com/en-us/library/ms178504(SQL.90).aspx

1.19.4 *Database Mail (Level 1, Scorable)*

Description:
Disable Database Mail where messaging is not required

Rationale:
Database Mail is a new component for sending e-mail messages from the Database Engine using SMTP. Enable Database Mail stored procedures only if you plan to configure and use Database Mail.

Remediation:
View the status of this feature in the Surface Area Configuration tool. Disable this feature if it is enabled and not required.

Audit:
To view the status of this option, perform the following steps:

1. Open the SQL Server Surface Area Configuration tool. This can be run from the SQL Server folder in the Start Menu or you can run the `SqlSAC.exe` tool from the SQL Server Program files installation path, such as: `C:\Program Files\Microsoft SQL Server\90\Shared\SqlSAC.exe`
2. Select the "Surface Area Configuration for Features" option under the Configure Surface Area options list.
3. Select the "Database Mail" option under the Database Engine dropdown list.

Default Value:

This feature is disabled by default in new installations of SQL Server.

References:
http://msdn.microsoft.com/en-us/library/ms188022(SQL.90).aspx

1.19.5 *Native XML Web Services (Level 1, Scorable)*

Description:
Do not configure XML Web Services endpoints where not required

Rationale:
Native XML Web Services provide database access over HTTP using SOAP messages. To use this capability you must create and start HTTP endpoints.

Remediation:
View the status of this feature in the Surface Area Configuration tool. Disable this feature if it is enabled and not required.

Audit:
To view the status of this option, perform the following steps:

1. Open the SQL Server Surface Area Configuration tool. This can be run from the SQL Server folder in the Start Menu or you can run the `SqlSAC.exe` tool from the SQL Server Program files installation path, such as: `C:\Program Files\Microsoft SQL Server\90\Shared\SqlSAC.exe`
2. Select the "Surface Area Configuration for Features" option under the Configure Surface Area options list.
3. Select the "Native XML Web Services" option under the Database Engine dropdown list.

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

References:
http://msdn.microsoft.com/en-us/library/ms188177(SQL.90).aspx

1.19.6 *Service Broker (Level 1, Scorable)*

Description:
Do not configure Service Broker endpoints where not required.

Rationale:
Microsoft SQL Server uses a Service Broker endpoint for Service Broker communication outside the instance of SQL Server. An endpoint is a SQL Server object that represents the capability for SQL Server to communicate over the network. Each endpoint supports a specific type of communication.

Remediation:
View the status of this feature in the Surface Area Configuration tool. Disable this feature if it is enabled and not required.

Audit:

To view the status of this option, perform the following steps:

1. Open the SQL Server Surface Area Configuration tool. This can be run from the SQL Server folder in the Start Menu or you can run the `SqlSAC.exe` tool from the SQL Server Program files installation path, such as: `C:\Program Files\Microsoft SQL Server\90\Shared\SqlSAC.exe`
2. Select the "Surface Area Configuration for Features" option under the Configure Surface Area options list.
3. Select the "Service Broker" option under the Database Engine dropdown list.

Default Value:

This feature is disabled by default in new installations of SQL Server. The SQL Server Endpoints needed for Service Broker to communicate between SQL Server instances are not configured by default.

References:
http://msdn.microsoft.com/en-us/library/ms180225(SQL.90).aspx

1.19.7 *Web Assistant (Level 1, Scorable)*

Description:
Disable Web Assistant where not required.

Rationale:
The Web Assistant stored procedures generate HTML files from Microsoft SQL Server data. In Microsoft SQL Server 2005, these stored procedures have been deprecated. Use this dialog to enable the procedures used by the Web Assistant.

Remediation:
View the status of this feature in the Surface Area Configuration tool. Disable this feature if it is enabled and not required.

Audit:
To view the status of this option, perform the following steps:

1. Open the SQL Server Surface Area Configuration tool. This can be run from the SQL Server folder in the Start Menu or you can run the `SqlSAC.exe` tool from the SQL Server Program files installation path, such as: `C:\Program Files\Microsoft SQL Server\90\Shared\SqlSAC.exe`

2. Select the "Surface Area Configuration for Features" option under the Configure Surface Area options list.
3. Select the "Web Assistant" option under the Database Engine dropdown list.

Default Value:
This feature is disabled by default in new installations of SQL Server.

References:
http://msdn.microsoft.com/en-us/library/ms188002(SQL.90).aspx

1.19.8 *Ad Hoc Data Mining (Level 1, Scorable)*

Description:
Disable ad hoc data mining queries where not required

Rationale:
The Data Mining Extensions (DMX) OPENROWSET function supports ad hoc queries using external providers. If your applications and scripts do not use this function, disable ad hoc data mining queries.

Remediation:
View the status of this feature in the Surface Area Configuration tool. Disable this feature if it is enabled and not required.

Audit:
To view the status of this option, perform the following steps:

1. Open the SQL Server Surface Area Configuration tool. This can be run from the SQL Server folder in the Start Menu or you can run the `SqlSAC.exe` tool from the SQL Server Program files installation path, such as: `C:\Program Files\Microsoft SQL Server\90\Shared\SqlSAC.exe`
2. Select the "Surface Area Configuration for Features" option under the Configure Surface Area options list.
3. Select the "Ad Hoc Data Mining" option under the Analysis Services dropdown list.

Default Value:
This feature is disabled by default in new installations of SQL Server.

References:
http://msdn.microsoft.com/en-us/library/ms178818(SQL.90).aspx

1.19.9 *Anonymous Connections (Level 1, Scorable)*

Description:
Disable anonymous connections to the Analysis Services where not required

Rationale:

Anonymous connections allow unauthenticated users to establish connections with Microsoft SQL Server 2005 Analysis Services. Unless your applications require unauthenticated users to connect to the instance, disable anonymous connections.

Remediation:
View the status of this feature in the Surface Area Configuration tool. Disable this feature if it is enabled and not required.

Audit:
To view the status of this option, perform the following steps:

1. Open the SQL Server Surface Area Configuration tool. This can be run from the SQL Server folder in the Start Menu or you can run the `SqlSAC.exe` tool from the SQL Server Program files installation path, such as: `C:\Program Files\Microsoft SQL Server\90\Shared\SqlSAC.exe`
2. Select the "Surface Area Configuration for Features" option under the Configure Surface Area options list.
3. Select the "Anonymous Connections" option under the Analysis Services dropdown list.

Default Value:
This feature is disabled by default in new installations of SQL Server.

References:
http://msdn.microsoft.com/en-us/library/ms190054(SQL.90).aspx

1.19.10 *Linked Objects (Level 1, Scorable)*

Description:
"Enable links To other instances" should be disabled where not required.

Rationale:
By default, Microsoft SQL Server 2005 Analysis Services disallows the creation of linked objects, such as linked dimensions and linked measure groups. In order to enable the linking of these objects both instances need to enable this feature.

Remediation:
View the status of this feature in the Surface Area Configuration tool. Disable this feature if it is enabled and not required.

Audit:
To view the status of this option, perform the following steps:

1. Open the SQL Server Surface Area Configuration tool. This can be run from the SQL Server folder in the Start Menu or you can run the `SqlSAC.exe` tool from the SQL Server Program files installation path, such as: `C:\Program Files\Microsoft SQL Server\90\Shared\SqlSAC.exe`

2. Select the "Surface Area Configuration for Features" option under the Configure Surface Area options list.
3. Select the "Linked Objects" option under the Analysis Services dropdown list.

Default Value:
This feature is disabled by default in new installations of SQL Server.

References:
http://msdn.microsoft.com/en-us/library/ms188509(SQL.90).aspx

1.19.11 *User-Defined Functions (Level 1, Scorable)*

Description:
Disable loading of user-defined COM functions where not required

Rationale:
Analysis Services allows you to load assemblies containing user-defined functions. These functions can be based on the common language runtime (CLR) or can be Component Object Model (COM) objects. Objects developed using the CLR can be secured using the CLR security model. COM objects do not support this security model. Unless your applications require user-defined COM objects, disable the loading of these objects.

Remediation:
View the status of this feature in the Surface Area Configuration tool. Disable this feature if it is enabled and not required.

Audit:
To view the status of this option, perform the following steps:

1. Open the SQL Server Surface Area Configuration tool. This can be run from the SQL Server folder in the Start Menu or you can run the `SqlSAC.exe` tool from the SQL Server Program files installation path, such as: `C:\Program Files\Microsoft SQL Server\90\Shared\SqlSAC.exe`
2. Select the "Surface Area Configuration for Features" option under the Configure Surface Area options list.
3. Select the "User-Defined Functions" option under the Analysis Services dropdown list.

Default Value:
This feature is disabled by default in new installations of SQL Server.

References:
http://msdn.microsoft.com/en-us/library/ms189372(SQL.90).aspx

1.19.12 *Scheduled Events and Report Delivery (Level 1, Scorable)*

Description:

Disable scheduled events and report delivery where not required.

Rationale:
In Reporting Services, you can schedule report delivery, report snapshots, and report cache expiration. If you do not require subscription functionality, or the ability to schedule reports to run on specific dates and times, you can disable scheduling and delivery processing. Note that if you disable this feature, reports can only be run on demand, and e-mail and shared folder delivery are not available.

Remediation:
View the status of this feature in the Surface Area Configuration tool. Disable this feature if it is enabled and not required.

Audit:
To view the status of this option, perform the following steps:

1. Open the SQL Server Surface Area Configuration tool. This can be run from the SQL Server folder in the Start Menu or you can run the `SqlSAC.exe` tool from the SQL Server Program files installation path, such as: `C:\Program Files\Microsoft SQL Server\90\Shared\SqlSAC.exe`
2. Select the "Surface Area Configuration for Features" option under the Configure Surface Area options list.
3. Select the "Scheduled Events and Report Delivery" option under the Reporting Services dropdown list.

Default Value:
This feature is enabled by default in new installations of SQL Server.

References:
http://msdn.microsoft.com/en-us/library/ms188468(SQL.90).aspx

1.19.13 *Web Service and HTTP Access (Level 1, Scorable)*

Description:
Disable Web Service and HTTP access where not required.

Rationale:
You can configure the Report Server web service to ignore all Simple Object Access Protocol (SOAP) requests and URL access requests. Disable HTTP and Web service requests if you do not have client applications that use the Web service, and if you do not require Report Manager, Report Builder, or Microsoft SQL Server Management Studio for this Reporting Services installation.

Remediation:
View the status of this feature in the Surface Area Configuration tool. Disable this feature if it is enabled and not required.

Audit:
To view the status of this option, perform the following steps:

1. Open the SQL Server Surface Area Configuration tool. This can be run from the SQL Server folder in the Start Menu or you can run the `SqlSAC.exe` tool from the SQL Server Program files installation path, such as: `C:\Program Files\Microsoft SQL Server\90\Shared\SqlSAC.exe`
2. Select the "Surface Area Configuration for Features" option under the Configure Surface Area options list.
3. Select the "Web Service and HTTP Access" option under the Reporting Services dropdown list.

Default Value:
This feature is enabled by default in new installations of SQL Server.

References:
http://msdn.microsoft.com/en-us/library/ms188941(SQL.90).aspx

1.19.14 *Windows Integrated Security (Level 1, Scorable)*

Description:
Enable Windows integrated security for report data source connections.

Rationale:
In Reporting Services, report data sources can use Windows integrated security to connect to external data sources. Using Windows integrated security, however, may allow access to data under a user's identity without the user's knowledge. To ensure that all report data source connections use an explicitly specified set of credentials, disable Windows integrated security for report data sources.

Remediation:
View the status of this feature in the Surface Area Configuration tool. Disable this feature if it is enabled and not required.

Audit:
To view the status of this option, perform the following steps:

1. Open the SQL Server Surface Area Configuration tool. This can be run from the SQL Server folder in the Start Menu or you can run the `SqlSAC.exe` tool from the SQL Server Program files installation path, such as: `C:\Program Files\Microsoft SQL Server\90\Shared\SqlSAC.exe`
2. Select the "Surface Area Configuration for Features" option under the Configure Surface Area options list.

3. Select the "Windows Integrated Security" option under the Reporting Services dropdown list.

Default Value:
This feature is enabled by default in new installations of SQL Server.

References:
http://msdn.microsoft.com/en-us/library/ms180255(SQL.90).aspx

Appendix A: References

1. Apple, Inc. (2008). *iPhone User Guide: For iPhone and iPhone 3G*. Available: http://manuals.info.apple.com/en_US/iPhone_User_Guide.pdf. Last accessed 27 March 2009.

2. Apple, Inc. (2008). *iPhone And iPod Touch Enterprise Deployment Guide*. Available: http://manuals.info.apple.com/en_US/Enterprise_Deployment_Guide.pdf. Last accessed 27 March 2009.

3. Jonathan Zdziarski (2008). *iPhone Forensics: Recovering Evidence, Personal Data, and Corporate Assets*. USA: O'Reilly.

4. National Institute of Standards and Technology. (2006). *NIST Special Publication 800-63: Electronic Authentication Guideline*. Available: http://csrc.nist.gov/publications/nistpubs/800-63/SP800-63V1_0_2.pdf. Last accessed 27 March 2009.

5. National Institute of Standards and Technology. (2008). *NIST Special Publication 800-124: Guidelines on Cell Phone and PDA Security*. Available: http://csrc.nist.gov/publications/nistpubs/800-124/SP800-124.pdf. Last accessed 27 March 2009.

Appendix B: Change History

Date	Version	Changes for this version
December 5, 2008	1.0.0	Public Release
December 5, 2008	1.1.0	Reference CIS SQL Server 2005 Benchmark v1.2.0 Change History here http://benchmarks.cisecurity.org/en-us/?route=downloads.show.single.sql2005.120
February 19, 2009	1.1.1	Reference CIS SQL Server 2005 Benchmark v1.2.0 Change History here http://benchmarks.cisecurity.org/en-us/?route=downloads.show.single.sql2005.120
February 12, 2010	1.2.0	Reference CIS SQL Server 2005 Benchmark v1.2.0 Change History here http://benchmarks.cisecurity.org/en-us/?route=downloads.show.single.sql2005.120
December 31, 2011	2.0.0	Public Release – Rewrote and reformatted benchmark. Resolved issues identified here.