

CERT-In

Indian Computer Emergency Response Team
Enhancing Cyber Security in India

Securing MS-SQL Server 2000

Department of Information Technology
Ministry of Communications and Information Technology
Govt. of India

Table of Contents

1. Introduction	3
2. Pre-Installation Settings	3
3. SQL Server Installation	4
4. Configuring SQL Server	4
4.1) Minimizing SQL Server	4
4.2) Configuring Permissions for SQL Server Objects	5
4.3) Configuring Permissions for Public	6
4.4) User Accounts	7
4.5) Auditing	11
5. Backup	14
6. Securing SQL Server Serving a Web Application	14
6.1) SQL Server Positioning	14
6.2) Securing the Network	15
6.3) Securing SQL Server From Attacks Through Web Applications	15
7. Audit and Penetration Tests	16
8. References	16

1. Introduction

A database server is a central repository of information of an organization. The more important the data for the organization, more tightly should it be guarded by using 'defense in depth' methodology. The principle of 'defense in depth' involves setting up multiple layers of protection mechanisms to safeguard the data.

This document is intended to act as a reference to help secure Microsoft SQL Server 2000 hosted on Windows 2000 Advanced Server.

All the commands and instructions have been tested on SQL Server 2000 with MS Windows 2000 as the operating system. The commands shown in the document require an account with Windows administrator privileges or a SQL Server account with 'sysadmin' privileges for execution unless otherwise stated.

2. Pre-Installation Settings

- 2.1) The underlying Operating System on which MS SQL server 2000 is installed should be hardened and unnecessary services should be stopped. Refer to CERT-In guidelines for securing MS Windows 2000 Advanced Server:
<http://www.cert-in.org.in/guidelines/CISG-2003-09.pdf>
- 2.2) The TCP/IP implementation in the Win2K server should be hardened to counter network DoS attacks. To harden the TCP/IP stack, refer to [Microsoft Corporation documentation](#).
- 2.3) Install the MS SQL Server on a dedicated server. Avoid installing the SQL Server on a machine hosting domain controller or a web server.
- 2.4) Install the SQL Server on a partition other than system partition.
- 2.5) The SQL server should always be installed on an NTFS partition to take advantages of the built-in security features and permission restrictions available in NTFS file system. To convert a partition to NTFS partition, use Windows 'convert' utility.

```
Cmd Prmpt> CONVERT volume /FS:NTFS [/V] [/CvtArea:filename  
[/NoSecurity] [/X]
```

- 2.6) Create an account with least privileges to run various SQL Server services instead of using an account with administrator privileges. The minimum privileges required for the service account are:
 - Read and Execute access on the SQL Server installation directory.
 - Full access to all .mdf, .ldf and .ndf database files.
 - Full access to SQL Server registry settings.

- 2.7) The database server system should be disconnected from the network till the installation is complete and setup files have been sanitized. Make sure proper access controls and logging mechanisms are in place before reconnecting the server back to the network.

3. SQL Server Installation

- 3.1) 'Custom Installation' should be preferred over 'Standard Installation'. 'Custom Installation' enables more secure features to be selected and installed.
- 3.2) Avoid installing the following components unless their functionality is required:
- Upgrade Tools
 - Code Samples
 - Books Online
 - Development Tools
 - Full Text Search

Unless Replication is required, 'Replication Support' should not be installed.

- 3.3) Disable Microsoft DTC if its functionality is not required.
- Run KillPwd utility to clean the setup files of any passwords that have been stored in clear text. It is available for download from [Microsoft Corporation](#).
 - Apply the latest service pack available. Apply all the patches and hot fixes available at Microsoft's website. To check the patch level:

```
Select @@version
```

Microsoft provides some tools to detect the missing service packs and patches. Microsoft Baseline Security Analyzer (MBSA) can be used to perform this task. It is a free tool available for download from [Microsoft Corporation](#). MBSA requires internet connectivity to download the latest security settings file. This file may also be downloaded manually and saved in the MBSA program directory.

4. Configuring SQL Server

4.1) Minimizing SQL Server

- In a default SQL Server installation, the program and data files are installed at '\Program Files\Microsoft SQL Server\' directory and at '\Program Files\Microsoft SQL Server\MSSQL\$InstanceName\' for Named Instance. The services installed by a default installation of SQL Server are:
 - MSSQLSERVER
 - MSQLErrorReporting
 - Microsoft Search
 - SQLSERVERAGENT

Other than MSSQLSERVER service, all other services are optional for SQL Server. Disable them unless specifically required. To disable the unnecessary services:

```
Run >> MMC >> File >> Select Services Or  
Control Panel >> Administrative Tools >> Services  
Select relevant service > Set 'Startup Type' = Disabled.
```

- Delete all unnecessary default databases like Pubs, Northwind etc. Do not delete system databases Master, msdb, model and tempdb.
- SQL Server supports a lot of protocols for connectivity. Select only the bare minimum protocols required. It is recommended that only TCP/IP protocol be selected. Disable all other protocols. To enable only selected protocol :

```
Start >> SQL Server Programs >> Server Network Utility
```

By using TCP/IP protocol, TCP/IP filtering and IPSec restrictions can be used to enforce the only ports on which the SQL Server will accept connections.

4.2) Configuring Permissions for SQL Server Objects

- One of the most important extended stored procedures, xp_cmdshell, gives access to underlying operating system commands. Xp_cmdshell should be dropped if its functionality is not required. To drop xp_cmdshell, use:

```
sp_dropextendedproc xp_cmdshell
```

If it is not feasible to drop this procedure, the access to this procedure should be restricted strictly to sysadmin only. Revoke access to xp_cmdshell from all other users. To restrict xp_cmdshell to sysadmin role only, use:

```
SQL Server Program >> Enterprise Manager >> Microsoft SQL Server  
>> SQL Server Group >> SQL Server Instance >> Management  
>> SQL Server Agent >> Properties >> Job System  
Check the 'Only users with SysAdmin privileges can execute  
CmdExec and ActiveScripting job steps' box.
```

The SQL Server needs to be restarted for the changes to take effect.
To check the user permissions on xp_cmdshell, use:

```
sp_helprotect xp_cmdshell
```

- SQL Server uses registry extended procedures to read, write and enumerate values and keys in the registry. These procedures include:

xp_regaddmultistring	xp_instance_regaddmultistring
xp_regdeletekey	xp_instance_regdeletekey
xp_regdeletevalue	xp_instance_regdeletevalue

xp_regenumvalues xp_regenumkeys xp_regread xp_regremovemultistring xp_regwrite	xp_instance_regenumkeys xp_instance_regenumvalues xp_instance_regread xp_instance_regremovemultistring xp_instance_regwrite
--	---

These procedures may be deleted from the master alongwith the DLL housing them, i.e. xpstar.dll. But these procedures should be dropped only if their functionality is not required. An application using these procedures will cease to work if these procedures are deleted. Also, applications like Enterprise Manager may not work properly if these procedures are dropped. To delete these procedures, use:

```
sp_dropextendedproc @funcname='Procedure_Name'
```

If it is not feasible to drop these procedures, it should be ensured that only sysadmin role has access to these procedures.

- If SQL Mail functionality is not required, consider dropping the following procedures:

xp_stopmail, xp_startmail, xp_deletemail, xp_sendmail

To drop these procedures, use:

```
sp_dropextendedproc @funcname='Procedure_Name'
```

If it is not feasible to drop these procedures, revoke access from all users for these procedures if SQL Mail is not being used.

4.3) Configuring Permissions for Public

- By default, the public group has permission to execute xp_regread which is used by SQL Server to read, write, and enumerate values and keys in the registry. All other registry extended stored procedures have execute permission granted only to dbo in master database by default. The execute permission for xp_regread from public should be removed by:

```
Use master
Revoke Execute On xp_regread From public
```

- By default, the public group has permission to create jobs in the database. If not required, this privilege should be revoked from the public group as it can be used to escalate the privilege level. It can be achieved by denying execute permission on following procedures:

Sp_add_job sp_add_jobstep sp_add_jobserver sp_start_job	xp_execresultset xp_printstatements xp_displayparamstmt
--	---

To remove the execute permissions from public group, use:

```
Use msdb / Use master
```

```
Revoke Execute On Procedure_Name From public
```

- By default, the public group has permission to modify data in the mswebtasks table. To revoke permission to INSERT, UPDATE, DELETE, or SELECT from public for the table mswebtasks, use:

```
Use msdb
Revoke All On mswebtasks From Public
```

- By default, the public group has Execute permission on sp_readwebtask. Web tasks may contain sensitive information. So, permission should be revoked from Public group as only administrators should be able to view the web tasks. To revoke access from public, use:

```
Use master
Revoke Execute On sp_readwebtask From Public
```

- If SQL Server Agent is configured to run using SQL Server Authentication, it stores the username and password in the registry. By default, public has permission to query this information from registry using sp_get_sqlagent_properties. To prevent public from accessing this information:

- Remove guest user from msdb database using:

```
Use msdb
Sp_DropUser 'guest'
```

- Revoke public permission from sp_get_sqlagent_properties using:

```
Use msdb
Revoke Execute On sp_get_sqlagent_properties From Public
```

- If DTS package is being used, the public group, by default, has privilege to execute the stored procedures sp_enum_dtspackages and sp_get_dtspackage. These procedures enable the public group to query the password used in the DTS package. To revoke access to these procedures by public, use:

```
Use msdb
Revoke Execute On sp_enum_dtspackages From Public
Revoke Execute On sp_get_dtspackage From Public
```

- By default, the public group in the master database has SELECT permissions on all columns in the syslogins table except the password hashes. These permissions should be revoked as only sysadmin role should have access to these columns. To revoke these permissions from public, use:

```
Use master
Revoke Select On syslogins From Public
```

- It is advisable not to assign any privileges to the 'public' role. Since 'public' permissions are inherited by all users, any privilege granted to 'public' is passed on to all the users.

4.4) User Accounts

- The account being used for running SQL Server service should be allocated least possible privileges. It should not be made a member of any administrative group or sysadmin role to minimize damages if the account gets compromised.
- Since SQL Server accounts are susceptible to brute force and dictionary password cracking attacks, the default accounts should be secured and strong password policy should be implemented.
- Accounts should not be shared among different people. This reduces accountability and increases chances of security breaches.

4.4.1. SQL Server Accounts

Check the following parameters

- login mode
- default login
- default domain
- audit level

These settings can be viewed using

```
Use master
xp_loginconfig
```

Inferences to be made on the above parameters

Login Mode-

SQL Server supports two types of authentication modes:

- Windows Authentication
- Mixed Mode Authentication

Windows authentication should be preferred over Mixed Mode authentication as it offers greater security and flexibility. Mixed mode authentication should be used only if specific requirements make it imperative to use it. However, since SQL Server doesn't support account lockout feature, SQL Server authentication is susceptible to brute force and dictionary password cracking attacks.

The SQL Authentication mode can be changed by

```
SQL Server Properties > Security tab > Check Windows Only > Apply
```

Default Login-

Ensure that default login in Windows authentication is not sa or a sysadmin.

Audit Level-

Recommended login audit level is 'All'. Minimum audit level for logins should be 'Failure'. To check the audit level set:

- Set a strong password for 'sa' login. In no case should the 'sa' be left with a blank password.
- It is recommended to remove Guest user from all the databases except master and tempdb, if its functionality is not required. To remove a guest user, use:

```
Use [Database_Name]
sp_dropuser 'guest'
```

- List all the logins of the database and check whether all the logins link to a genuine user. To list all the logins, use:

```
Use master
Select * From sysxlogins
```

Ensure that there are no dummy or suspicious accounts. Also, ensure that there is no guest user in any database except master and msdb.

- List all the windows authenticated users. Ensure that they are trusted users and need access to the SQL Server. To list all windows authenticated users, use:

```
Use master
Select name, loginname From syslogins Where isntname=1
```

- List all the users having access to master database. Ensure that only trusted users have access to master. To list all users having access to master database, use:

```
Use master
Select Name From sysxlogins Where dbid=1
```

- No user should be allowed to have a blank password. The password policy of the organization should be implemented strictly. To list all the users having a blank password, use:

```
Use master
Select name From sysxlogins Where password Is NULL
```

- It should be ensured that there is no windows account in SQL Server which has been deleted from the Windows. To check for such accounts, use:

```
Use master
Sp_validatelogins
```

- SQL Server provides some Fixed Server Roles and Fixed Database Roles. It should be ensured that members of these roles actually require these privileges. Any user who does not require these privileges should be removed from these roles.

- To list the users belonging to Fixed Server Roles, use:

```
Use master
Select name, loginname From syslogins Where Srvr_Role_Name=1
```

- To list users belonging to Fixed Database Roles, use:

```
Use Database_Name
sp_helprolemember Database_Role_Name
```

- To view the privileges granted to a particular user, use:

```
Sp_helpprotect 'User_Name'
```

- SQL Server allows a login to access objects in different databases by using database permissions chaining. This permission chaining can be exploited to allow a user in the db_owner group of a less significant database to gain full control over a server. The database permission chaining should be disabled, if not required, by:

```
Enterprise Manager >> Tools >> 'SQL Server Config. Properties
'Security' >> Uncheck 'Allow cross-database ownership chaini
```

- Check the attributes of all the databases and ensure that they belong to trusted users. To check these attributes:

```
Use master
Sp_helppdb
```

4.4.2.Windows Accounts and access permissions

- By default, Domain Administrators are members of 'sysadmin' role and have full access over SQL server. If Domain Administrator should not have allowed full rights over SQL Server, then the Windows local BUILTIN/Administrator group should be removed from the 'sysadmin' role. To remove BUILTIN/Administrator from 'sysadmin' role, use:

```
SQL Server Program >> Enterprise Manager >> Microsoft SQL Server
>> SQL Server Group >> SQL Server Instance >> Security >> Logins
>> BUILTIN/Administrator >> Right Click >> Delete.
```

- The Windows 'guest' account should be disabled to prevent remote access to SQL Server through 'guest' user. Also, Null sessions should be disabled to prevent unauthenticated users from logging on to the server anonymously. To disable Null sessions, use:

```
Run >> Regedit
Select key..
HKLM\System\CurrentControlSet\Control\LSA\RestrictAnonymous
Set RestrictAnonymous value = 1.
```

- Access to the SQL Server system from network should be restricted to trusted users only. 'Everyone' group should not be allowed to access the SQL Server system from network. To disable 'Everyone' group from accessing the SQL Server system, use:

```
Control Panel >> Administrative Tools >> Local Security Settings
```

```
>> Local Policies >> User Rights Assignment  
Remove 'Everyone' from 'Access this computer from the network' setting.
```

- Ensure that 'Everyone' does not has permissions on the following registry keys:
 - Default instance:
HKLM\Software\Microsoft\MSSQLServer
 - Named Instance
HKLM\Software\Microsoft\Microsoft SQL Server\InstanceName
 - SQL Server Service

HKLM\System\CurrentControlSet\Services\MSSQLSERVER

To check the privileges for these keys, use:

```
Start >> Run >> Regedit
```

- Ensure that 'Everyone' does not have permissions to the SQL Server files and directories. The default SQL Server directory is '\Program Files\Microsoft SQL Server\MSSQL\'. It may be different for custom installation.
- The SQL Server Service account should be checked to ensure that it has not been granted excessive privileges. The SQL Server service should have only the following permissions:
 - Install Location: Read and Execute only.
 - Database File Directory : All Permissions
 - Error Log File Directory : All Permissions
 - Backup File Directory : All Permissions
 - Job Temporary File Output Directory : All Permissions

- The SQL Server should not have any shares. In case shares are required, ensure that only specific permissions are granted. 'Everyone' should not be given access to shares. To view shares, use:

```
Control Panel >> Administrative Tools >> Computer Management  
>> Shared Folders >> Shares
```

Remove any unnecessary shares.

4.5) Auditing

Auditing does not prevent attacks from occurring but can be very helpful in tracing the footprints of the attacker and the attack methodology.

- Implement auditing at both Operating System level and SQL Server level.
- Audit logs should be saved on the disk instead of saving in database. In case of SQL Server compromise, the attacker will have access to audit logs if they are saved in the database.

- It is strongly recommended that database audit logs be saved on a centralized syslog server. To implement a Centralized Syslog Server, refer to CERT-In Guidelines for setting up a Syslog Server:

<http://www.cert-in.org.in/guidelines/CISG-2004-03.pdf>

4.5.1. Auditing at Operating System Level

It is recommended to enable auditing of all actions across the Operating System. At a minimum, auditing on the following events should be enabled:

- Failed Login attempts.

To enable, use:

```
Control Panel >> Administrative Tools >> Local Security Settings  
Local Policies >> Audit Policy >> 'Audit Account Logon Events'  
Check 'Failure' checkbox.  
Local Policies >> Audit Policy >> 'Audit Logon Events'  
Check 'Failure' checkbox.
```

- Attempts to Change User Rights, Audit Policies and Trust Relationships. To enable, use:

```
Control Panel >> Administrative Tools >> Local Security Settings  
Local Policies >> Audit Policy >> 'Audit Policy Change'  
Check both 'Success' and 'Failure' checkboxes.
```

- Attempts to perform an action that may affect the security of the system or the system logs. To enable, use:

```
Control Panel >> Administrative Tools >> Local Security Settings  
Local Policies >> Audit Policy >> 'Audit System Events'  
Check both 'Success' and 'Failure' checkboxes.
```

- Attempts to perform an action that requires more privileges than the privileges assigned to the user. To enable, use:

```
Control Panel >> Administrative Tools >> Local Security Settings  
Local Policies >> Audit Policy >> 'Audit Privilege Use'  
Check 'Failure' checkboxes.
```

- Attempts to access objects in the system like files, registry etc for which the user does not has privilege. To enable, use:

```
Control Panel >> Administrative Tools >> Local Security Settings  
Local Policies >> Audit Policy >> 'Audit Object Access'  
Check 'Failure' checkboxes.
```

Select the file structure level at which the auditing is to be enabled.

```
Right Click >> Properties >> Security >> Advanced >> Auditing  
Add >> Select 'Everyone' >> Select 'OK' >> Select All Events  
>> Apply
```

4.5.2. Auditing at SQL Server Level

Auditing for logins is not enabled in the SQL Server by default. It is recommended that all logins should be audited. At a minimum, auditing should record all failed login attempts.

- To enable auditing in SQL Server, use:

```
L Server Program >> Enterprise Manager >> Microsoft SQL Server  
SQL Server Group >> SQL Server Instance >> Properties  
Security >> 'Audit Level' >> Check 'All' checkbox.
```

The SQL Server needs to be restarted for the setting to take effect.

- SQL Server meets the C2 auditing level. It is implemented in Profiler utility. By default, C2 auditing is not enabled in SQL Server. To enable C2 level auditing, use:

```
L Server Programs >> Profiler >> New >> Trace
```

Select the events on which to perform auditing and apply filters. This trace can be saved as a template for future use.

- It is strongly recommended that audit logs from Profiler be saved to a file on the disk or on a Syslog server rather than in a table. Storing the logs in a table will degrade the SQL Server performance.

Checking important SQL Server Parameters

- Check all the jobs listed in SQL Server Agent that are scheduled to take place at a fixed time. These jobs should be checked to ensure that they have been created by genuine users and no malicious activity is taking place. To check for scheduled jobs:

```
Use msdb  
sp_help_job
```

- To check the server configurations, use:

```
Use master  
sp_configure
```

Main configuration parameters to look for are:

- 'C2 Audit Mode' – Should be enabled. Value = 1
 - 'Remote Access' – Should be disabled. Value = 0
 - 'Scan for startup procs' - If enabled, check the procedures which are executed on startup for any malicious code.
 - 'Cross DB Ownership Chaining' – Disable if Database Ownership Chaining is not required.
- In case of database server storing sensitive information, 'C2 Audit Mode' should be enabled. If the database server is unable to write to the audit file for any reasons, 'C2 Audit Mode' forces the database to stop. To enable 'C2' Auditing:

```
USE master
EXEC sp_configure 'show advanced option', '1'
RECONFIGURE
sp_configure 'c2 audit mode', 1
```

The SQL Server needs to be restarted for the tracing to start.

- By default, SQL Server writes error logs to Program Files\Microsoft SQL Server\Mssql\Log directory. The maximum number of error log files should be set to a value high enough so that the error logs are not overwritten. Error logs should be overwritten only after being archived or properly reviewed. To set the maximum number of error log files to 25000 or greater:

- For default instance:

```
Set NumErrorLogs in the registry key
HKEY_LOCAL_MACHINE\Software\MSSQLServer\MSSQLServer to 25000
or greater.
```

- For named instance:

```
Set NumErrorLogs in the registry key
HKEY_LOCAL_MACHINE\Software\Microsoft\Microsoft SQL
Server\[Instancename]\MSSQLServer to 25000 or greater.
```

This key should be created as REG_DWORD type if not already present.

5. Backup

- Create and implement a backup policy for the SQL Server.
- Backup all the system databases like master and msdb regularly.
- All user databases should be backed up in accordance with the organizations backup policy. Implement a combination of differential and complete backup.
- As a security precaution, save the backups on a remote system or on an external storage device like tape drive. These storage devices should be kept in a secure location with restricted access.
- Secure the backup with password protection feature of SQL Server.
- Restrict access to the backup files to most trusted users only.

6. Securing SQL Server Serving a Web Application

6.1) SQL Server Positioning

- Never expose SQL Server directly to outside world.

- SQL Server should not be placed in the DMZ. Secure the SQL Server in the Militarized Zone behind a second firewall.
- Do not host the SQL Server on the same system as web server or application server.
- Access to SQL Server should be restricted to only the web server or the application server.
- For more information on securing perimeter security, refer to [CERT-In guidelines for Securing Perimeter Defence](#).

6.2) Securing the Network

- Block SQL Server ports on perimeter firewall. All packets routed for SQL Server from outside the perimeter should be blocked at the perimeter firewall itself.
- Block any traffic going out from SQL Server on any unusual port like TCP 21, 80, 139 or UDP 53.
- Use SSL for protocol encryption of all server network libraries. To enable protocol encryption on server side, use:

```
SQL Server Programs >> Server Network Utility >> General  
>> Force Protocol Encryption
```

To enable protocol encryption on client side, use:

```
SQL Server Programs >> Client Network Utility >> General  
>> Force Protocol Encryption
```

Implementing SSL requires a Certificate issued by a Certifying Authority.

6.3) Securing SQL Server From Attacks Through Web Applications

- Web Applications should be run with least possible privilege level.
- Ensure strong passwords for all the accounts.
- Any input being received by the web application should be comprehensively checked for any malicious data.
- Reject any input received from web application which is known to be bad. If the input being expected is numeric, then any string input should be rejected.

- Allow the input which meets the validation criterion after checking it for any unintended values. Filter out characters like ‘, ; -- etc and keywords like ‘Select’, ‘Delete’, ‘Union’, ‘Exec’, ‘xp_cmdshell’, ‘Shutdown’ etc.
- Restrict the access of web applications to stored procedures as far as possible. Avoid giving direct access to tables and views to the web application.
- Implement stored procedures using ADO command object. This enhances the security using strongly typed variables.
- Do not allow users to run SQL queries directly on the server.
- Do not allow ad-hoc basis through OLEDB from SQL Server unless this functionality is specifically required. To disable ad-hoc queries, use:
Start >> Run >> Regedit
 - For default instance:
HKLM\Software\MSSQLServer\MSSQLServer\Providers
Set DisallowAdhocAccess value = 1
 - For named instance:
HKLM\Software\Microsoft\Microsoft SQL Server\[Instancename]\Providers
Set DisallowAdhocAccess value = 1
- Encourage code review among developers and peer testing of the code.
- Suppress default error banners. Replace them with custom error banners to ensure that attackers can't glean any information from them

7. Audit and Penetration Tests

- The SQL Server should be subjected to regular audit and penetration tests. This helps in keeping the security configurations inline with security best practices.
- Microsoft provides a useful tool, ‘Best Practices Analyzer’ (MBPA), for SQL Server to check for the compliance of SQL Server installation with the best practices for SQL Server operation and management. This is a free tool available for download from Microsoft:

<http://www.microsoft.com/downloads/details.aspx?FamilyID=b352eb1f-d3ca-44ee-893e-9e07339c1f22&DisplayLang=en>

- Various third party tools, free as well as paid, are available to check the security configurations of the SQL Server. These tools may be used to perform the audit of the SQL Server.

8. References

- <http://www.microsoft.com/technet/prodtechnol/sql/2000/books/pkadmindefault.msp#>
- <http://www.microsoft.com/technet/security/prodtech/dbsql/sql2kaud.msp#>
- <http://www.sans.org/rr/papers/index.php?id=9>
- <http://www.cert-in.org.in/presentation/29thjuly04/Database%20Server%20Security%20Overview.pdf>
- <http://www.microsoft.com/technet/prodtechnol/sql/70/maintain/secure.msp#>