

---

## Microsoft SQL Server 2000 Security

The following is a checklist of security guidelines to follow in securing Microsoft SQL Server 2000. It is divided into categories: 1) Setup and General Issues, 2) Authentication, Users and Logins, 3) Permissions, 4) Logging, and 5) Applications. Within each category, the guidelines are described as Mandatory, Recommended or Paranoid; these labels are intended to help you prioritize your actions and to weigh the relative importance of each guideline. The guidelines are intended for database administrators with some amount of experience, so step-by-step instructions are not included.

### Setup and General Issues

- **[Mandatory]** The Internet firewall should block all packets going to or coming from the database servers. Consider regulating traffic from the internal LAN as well. SQL Server primarily uses TCP 1433 for data sessions and UDP 1434 for pre-session handshaking. But be aware that SQL Server can also use UDP 137, TCP 139 and TCP 135.
- **[Mandatory]** Stay abreast of the latest Service Packs and hotfixes, and install them as soon as feasible; however, do not blindly install Service Packs or hotfixes without first testing them on non-production servers, and without first backing up the production server. Keep in mind that SQL Server has Service Packs separate from the operating system's Service Packs.
- **[Mandatory]** Physically secure database servers and their backup media in locked, access-controlled rooms. Provide electronic surveillance when appropriate.
- **[Mandatory]** In general, disable services or features of SQL Server that you are not currently using, such as SQL Mail.
- **[Mandatory]** Scan the database server with the Microsoft Baseline Security Analyzer (MBSA) or similar auditing tool to reveal common insecurities.
- **[Mandatory]** Regularly audit all shared folders on the database server to ensure that all permissions are minimal and that none of the shares are unnecessary. Be especially wary of shared folders that permit Write access.
- **[Mandatory]** When replicating databases over the Internet, set up a router-to-router Virtual Private Networking (VPN) tunnel to connect the two LANs. When replicating over untrusted internal networks, use IPSec to encrypt the replication traffic.
- **[Mandatory]** Assign a long and complex password to the SQL Server service account (preferably 25 characters or more, with 15 characters the minimum).

- **[Recommended]** When an Internet-accessible web server is a front end to a database, do not locate the database on the IIS server itself. Database server(s) should be located behind the firewall, perhaps inside the LAN. This step is borderline mandatory, but not always practical.
- **[Recommended]** It is almost always better (if not required) to make the database server a member of an Active Directory domain than to make it a stand-alone system. This ties into the benefits of Integrated Windows authentication, of course, but there are many other benefits as well, such as management through Group Policy. Group Policy should be used to enforce other commonsense security precautions, such as disabling null user session access and renaming the built-in Administrator account.
- **[Recommended]** In general, the SQL Server service account should not be the local System account. It is permissible (and often necessary) to use a global account for the SQL Server service account, but don't add it to the Domain Admins group if it requires administrative privileges-- simply add the account to the local Administrators group on each machine where the elevated privileges are required (usually on just the SQL Servers). Be aware that Kerberos authentication may fail if the Service Principal Name (SPN) is not correct on the SQL Server and in DNS (see the SETSPN.EXE tool from the *Resource Kit*).
- **[Recommended]** Install an anti-virus scanner on your database server, but exclude the folders which contain database files, transaction logs, snapshot files, or other similar files that are unlikely to be infected (and which would entail significant performance penalties if frequently scanned).
- **[Recommended]** Consider dropping the Northwind and Pubs databases.
- **[Recommended]** The Model database is copied whenever a new database is created. Consider hardening the settings in the Model database, such as removing the Guest login, so that all new databases will start with more secure defaults.
- **[Recommended]** Don't forget that the Encrypting File System (EFS) can be used to encrypt SQL Server and MSDE databases. In rare cases, such as on laptops or tradeshow servers, where the physical security of the server is at risk and peak performance is not terribly important, use EFS to encrypt the database files. (Note: you must encrypt these files while logged on as the SQL Server service account, and this account cannot be System.)
- **[Recommended]** Encrypt, or at least password protect, backup media containing sensitive data. This is especially true when storing media off-site.
- **[Paranoid]** Prevent the database server from appearing in Query Analyzer by using the "net.exe config server" command with the "/hidden:yes" switch.

## Authentication, Users and Logins

- **[Mandatory]** Assign a long and complex password to the built-in sa login, even when Integrated Windows authentication is being used (as a fail-safe).
- **[Mandatory]** Regularly audit the membership of the sysadmin role and the passwords of all accounts and logins in it.
- **[Mandatory]** When using Mixed mode authentication, assign long and complex passwords to all logins, especially the logins with dangerous permissions and roles. Never use blank or easy-to-guess passwords, such as "password".
- **[Mandatory]** Promptly remove unneeded logins and permissions, especially when potentially dangerous users are fired or dismissed.
- **[Mandatory]** Disable the built-in local Guest account in the operating system and assign it a long and random password. Do the same for the global Guest account in Active Directory. These Guest accounts exist even if SQL Server is not installed anywhere.
- **[Recommended]** Whenever possible, configure the server to use Integrated Windows authentication instead of the older Mixed authentication mode. (This is a different distinction than the one between "native mode" and "mixed mode" Active Directory domains.)
- **[Recommended]** When using Mixed mode authentication, it is still better to never use the sa login. Instead, create a Windows group, add the Windows user accounts of DBAs to it, create a SQL Server login for this group, and add the login to the sysadmin fixed server role.
- **[Recommended]** When using Mixed mode authentication, modify the sp\_password stored procedure to enforce restrictions on the length and complexity of new passwords. At the same time, regularly check the code of this procedure to make sure no one has modified it to collect passwords, i.e., that it hasn't been Trojaned.
- **[Recommended]** Use the -T switch with BCP.EXE and the -E switch with OSQL.EXE to use "trusted" Integrated Windows authentication.

## Permissions

- **[Mandatory]** Remove the Guest user from all databases except the master and tempdb databases, which require Guest access. This is the Guest user which is internal to SQL Server itself, not the Guest account in Active Directory or the local operating system.
- **[Mandatory]** Format all drive volumes on the database server with NTFS.

- **[Mandatory]** The setup program for SQL Server should have done this already, but set the NTFS permissions on all database-related files, including the SQL Server binaries, to permit only Read & Execute to Authenticated Users, and Full Control for System, Administrators and the SQL Server service account. If you change the SQL Server service account, do it using Enterprise Manager so that permissions will be changed automatically.
- **[Mandatory]** Use NTFS audit settings (SACLs) to track failed access to database-related files.
- **[Mandatory]** The setup program for SQL Server should have done this already, but, on the following registry keys, remove all permissions for the Everyone group and grant Full Control to the SQL Server service account and local Administrators, then audit all failed access and all successful writes to them as well: HKLM\Software\Microsoft\MSSQLServer (for a default instance) or HKLM\Software\Microsoft\Microsoft SQL Server\*Instancename* (for a named instance, if any).
- **[Mandatory]** Delegate authority over the server and its databases by utilizing fixed server and database roles appropriately, or by creating your own custom roles, so that excessive power is not being placed in the hands of those who don't need it. In short, do not simply place everyone in the sysadmin role out of convenience.
- **[Mandatory]** All database and server management scripts should be stored in an access-controlled NTFS folder. Audit all access. Scripts should not contain hard-coded passwords. Consider digitally signing these scripts using the Windows Script Host (WSH) 5.6 or later, and warn on the attempted execution of scripts that fail signature verification.
- **[Recommended]** Disable cross-database ownership chaining (KB810474). It is disabled by default after SP3 is applied to SQL Server 2000.
- **[Recommended]** Regularly audit what the Public role has access to. This role is roughly equivalent to the Everyone group in the operating system. Remove or limit the rights of the Public role whenever possible, but do not explicitly deny permissions for it (simply remove them).
- **[Recommended]** Regularly audit who has Execute permission on all stored procedures, especially xp\_cmdshell. The Guest and Public logins rarely need the Execute permission. When in doubt, grant permission only to the sysadmin role.
- **[Recommended]** When using UDL files, assign restrictive NTFS permissions and audit all access to the files. Avoid storing passwords in UDL files by using Integrated Windows authentication.

- **[Paranoid]** Consider removing the local Administrators group from the sysadmin role and replacing it with a custom local group with only the true database administrators. This will not prevent local Administrators from granting themselves any access they wish, but at least these actions would be auditable.

## Logging

- **[Mandatory]** Enable logging of authentications to the server at the operating system level using the Event Log, and expand the size of the Security log to at least 100MB.
- **[Mandatory]** Using the SQL Profiler, audit login events, password changes, role changes, and other classes of events that would be useful in your environment (without causing an unacceptable performance penalty).

## Applications

- **[Mandatory]** Never permit user applications to send arbitrary SQL commands to the server without performing some form of validation of the commands first, and never permit public Internet-accessible applications (such as web applications running on IIS) to send user-defined SQL commands to the back-end database, even with input validation rules.
- **[Recommended]** Whenever possible, user applications should be designed to call stored procedures, custom functions and views on the database server instead of permitting these applications to directly access the underlying tables themselves; stored procedures, functions and views should, in this sense, mediate and regulate all interaction with the database to validate requests and to hide the details of the database's design from the user application. Extensive use of stored procedures and server-side functions should also improve the performance of database applications.
- **[Recommended]** Sensitive data, such as Social Security and credit card numbers, should be encrypted when stored in the database and when transmitted over the network. Per-column encryption can be implemented with third-party toolsets or custom programming using the CryptoAPI interface.
- **[Recommended]** Disable any unused network libraries. Microsoft is moving towards making the TCP/IP net library the preferred one, hence, if you have a choice in the design of your client applications, design them around the TCP/IP net library.
- **[Recommended]** Consider using SSL for encrypting application traffic between clients and database servers, or using IPSec for encrypting all communications with the server whatsoever.

- **[Recommended]** Require IPSec AH or ESP for the channel between your front-end IIS servers on the Internet and your back-end SQL Server(s). Your database servers should drop all non-IPSec traffic. Install "IPSec-offload" network adapter cards on the servers to alleviate the performance penalties of using IPSec.
- **[Recommended]** Avoid hard coding passwords into connection strings in database applications. This will be much easier to do if Integrated Windows authentication is being used.
- **[Paranoid]** Encrypt the code of stored procedures, triggers and views using the "WITH ENCRYPTION" clause when creating them. This step goes into the mandatory category if you are selling turn-key database solutions or when you fear physical compromise of the server or its backup media. Don't forget to export a backup of the cleartext code first. Be aware, however, that easy-to-use tools exist to break the encryption.