**10 Steps to Help Secure SQL Server**

**Note:** For information on protecting your server against the Slammer worm, visit the [Finding and Fixing Slammer Vulnerabilities](http://www.microsoft.com/sql/prodinfo/previousversions/slammer.mspx) page. For detailed information about security features and best practices for SQL Server 2000 Service Pack 3 (SP3), read the article ["Microsoft SQL Server 2000 SP3 Security Features and Best Practices"](http://www.microsoft.com/technet/prodtechnol/sql/2000/maintain/sp3sec00.mspx) on the Microsoft TechNet site.

Here are ten things you can do today to improve the security of your SQL Server installation:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|

|  |  |
| --- | --- |
| Install the most recent service pack. | **Install the most recent service pack.**The single most effective action you can take to improve the security of your server is to upgrade to SQL Server 2000 Service Pack 4 (SP4). To download SP4, visit the [SQL Server 2000 SP4 page](http://www.microsoft.com/sql/prodinfo/previousversions/downloads.mspx). You should also install all security updates as they are released. To sign up to be notified by e-mail of new security updates, visit the [Product Security Notification page](http://www.microsoft.com/technet/security/bulletin/notify.mspx%20).  |

|  |  |
| --- | --- |
| Assess your server's security with Microsoft Baseline Security Analyzer (MBSA). | **Assess your server's security with Microsoft Baseline Security Analyzer (MBSA).**MBSA is a tool that scans for common insecure configurations in several Microsoft products including SQL Server and Microsoft SQL Server 2000 Desktop Engine (MSDE 2000). It can be run locally or over a network. It tests SQL Server installations for problems such as: Too many members of the **sysadmin** fixed server role. Granting of right to create CmdExec jobs to roles other than **sysadmin**. Blank or trivial passwords. Weak authentication mode. Excessive rights granted to the Administrators group. Incorrect access control lists (ACLs) on SQL Server data directories. Plaintext **sa** password in setup files. Excessive rights granted to the **guest** account. SQL Server running on a system that is also a domain controller. Improper configuration of the Everyone group, providing access to certain registry keys. Improper configuration of SQL Server service accounts. Missing service packs and security updates. Microsoft distributes MBSA as a free download. For complete documentation and the latest version of MBSA, visit the [MBSA page](http://www.microsoft.com/technet/security/tools/mbsahome.mspx).  |

|  |  |
| --- | --- |
| Use Windows Authentication Mode. | **Use Windows Authentication Mode.**Whenever possible, you should require Windows Authentication Mode for connections to SQL Server. This will shield your SQL Server installation from most Internet-based attacks by restricting connections to Microsoft Windows user and domain user accounts. Your server will also benefit from Windows security enforcement mechanisms such as stronger authentication protocols and mandatory password complexity and expiration. Also, credentials delegation (the ability to bridge credentials across multiple servers) is only available in Windows Authentication Mode. On the client side, Windows Authentication Mode eliminates the need to store passwords, which is a major vulnerability in applications that use standard SQL Server logins. To set up Windows Authentication Mode security with Enterprise Manager in SQL Server: Expand a server group. Right-click a server, and then click **Properties**. On the **Security** tab, under **Authentication**, click **Windows only**. For more information, see the "Authentication Mode" topic in SQL Server Books Online or on [MSDN](http://msdn.microsoft.com/library/en-us/instsql/in_runsetup_6p9v.asp?frame=true).  |

|  |  |
| --- | --- |
| Isolate your server and back it up regularly. | **Isolate your server and back it up regularly.**Physical and logical isolation make up the foundation of SQL Server security. Machines hosting a database should be in a physically protected location, ideally a locked machine room with monitored flood detection and fire detection/suppression systems. Databases should be installed in the secure zone of your corporate intranet and never directly connected to the Internet. Back up all data regularly and store copies in a secure off-site location. For guidance on backup procedures and other operational best practices, refer to the [SQL Server 2000 Operations Guide](http://www.microsoft.com/technet/prodtechnol/sql/2000/maintain/sqlops0.mspx).  |

|  |  |
| --- | --- |
| Assign a strong sa password. | **Assign a strong sa password.**The **sa** account should always have a strong password, even on servers that are configured to require Windows Authentication. This will ensure that a blank or weak **sa** password is not exposed in the future if the server is reconfigured for Mixed Mode Authentication. To assign the **sa** password: Expand a server group, and then expand a server. Expand **Security**, and then click **Logins**. In the details pane, right-click **SA**, and then click **Properties**. In the **Password** box, type the new password. For more information, see the "System Administrator (sa) Login" topic in SQL Server Books Online or on [MSDN](http://msdn.microsoft.com/library/en-us/adminsql/ad_security_8soe.asp?frame=true).  |

|  |  |
| --- | --- |
| Limit privilege level of SQL Server Services. | **Limit privilege level of SQL Server Services.**SQL Server 2000 and SQL Server Agent run as Windows services. Each service must be associated with a Windows account, from which it derives its security context. SQL Server allows users of the **sa** login, and in some cases other users, to access operating system features. These operating system calls are made with the security context of the account that owns the server process. If the server is cracked, these operating system calls may be used to extend the attack to any other resource to which the owning process (the SQL Server service account) has access. For this reason, it is important to grant only necessary privileges to SQL Server services. The following settings are recommended: **SQL Server Engine/MSSQLServer**If there are named instances, they will be named MSSQL$InstanceName. Run as a Windows domain user account with regular user privileges. Do not run as local system, local administrator, or domain administrator accounts. **SQL Server Agent Service/SQLServerAgent** Disable if not required in your environment; otherwise run as a Windows domain user account with regular user privileges. Do not run as local system, local administrator, or domain administrator accounts. **Important:** SQL Server Agent will need local Windows administrator privileges if one of the following is true: SQL Server Agent connects to SQL Server using standard SQL Server Authentication (not recommended). >SQL Server Agent uses a multiserver administration master server (MSX) account that connects using standard SQL Server Authentication. SQL Server Agent runs Microsoft ActiveX® script or CmdExec jobs owned by users who are not members of the **sysadmin** fixed server role. If you need to change the account associated with a SQL Server service, use SQL Server Enterprise Manager. Enterprise Manager will set appropriate permissions on the files and registry keys used by SQL Server. Never use the Services applet of Microsoft Management Console (in Control Panel) to change these accounts, because this requires manual adjustment of dozens of registry and NTFS file system permissions and Microsoft Windows user rights. For more information, see the Microsoft Knowledge Base article [Change the SQL Server Service Account Without Using SQL Enterprise Manager in SQL Server 2000](http://support.microsoft.com/default.aspx?scid=kb;en-us;Q283811). Changes to account information will take effect the next time the service is started. If you need to change the account associated with SQL Server and SQL Server Agent, you must apply the change to both services separately using Enterprise Manager.  |

|  |  |
| --- | --- |
| Disable SQL Server ports on your firewall. | **Disable SQL Server ports on your firewall.**Default installations of SQL Server monitor TCP port 1433 and UDP port 1434. Configure your firewall to filter out packets addressed to these ports. Additional ports associated with named instances should also be blocked at the firewall. |

|  |  |
| --- | --- |
| Use the most secure file system. | **Use the most secure file system.**NTFS is the preferred file system for installations of SQL Server. It is more stable and recoverable than FAT file systems, and enables security options such as file and directory ACLs and file encryption (EFS). During installation, SQL Server will set appropriate ACLs on registry keys and files if it detects NTFS. These permissions should not be changed. With EFS, database files are encrypted under the identity of the account running SQL Server. Only this account can decrypt the files. If you need to change the account that runs SQL Server, you should first decrypt the files under the old account, then re-encrypt them under the new account.  |

|  |  |
| --- | --- |
| Delete or secure old setup files. | **Delete or secure old setup files.**SQL Server setup files may contain plain-text or weakly encrypted credentials and other sensitive configuration information that has been logged during installation. The location of these log files varies depending on which version of SQL Server has been installed. In SQL Server 2000, the following files may be affected: sqlstp.log, sqlsp.log, and setup.iss in the <systemdrive>:\Program Files\Microsoft SQL Server\MSSQL\Install folder for a default installation, and the <systemdrive>:\Program Files\Microsoft SQL Server\ MSSQL$<Instance Name>\Install folder for named instances. If the current system is an upgrade from SQL Server version 7.0 installations, the following files should be checked as well: setup.iss in the %Windir% folder, and sqlsp.log in the Windows Temp folder. Microsoft distributes a free utility, Killpwd, which will locate and remove these passwords from your system. To learn more about this free download, see the Microsoft Knowledge Base article [Service Pack Installation May Save Standard Security Password in File](http://support.microsoft.com/default.aspx?scid=KB;en-us;263968&).  |

|  |  |
| --- | --- |
| Audit connections to SQL Server. | **Audit connections to SQL Server.**SQL Server can log event information for review by the system administrator. At a minimum, you should log failed connection attempts to SQL Server and review the log regularly. When possible, save these logs to a different hard drive than the one on which data files are stored. To enable auditing of failed connections with Enterprise Manager in SQL Server: Expand a server group. Right-click a server, and then click **Properties**. On the **Security** tab, under **Audit Level**, click **Failure**. You must stop and restart the server for this setting to take effect. For more information, see "[SQL Server 2000 Auditing](http://www.microsoft.com/technet/security/prodtech/sqlserver/sql2kaud.mspx)" on Microsoft TechNet and the "Using Audit Logs" topic in [SQL Server Books Online](http://www.microsoft.com/sql/techinfo/books.mspx) or on [MSDN](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/adminsql/ad_security_1yr7.asp).  |

 |