

informatizate

<http://www.informatizate.net>

MODELO DE SEGURIDAD DE SQL SERVER 7.0/2000

Robert G. Ancajima Rojas

Area de Desarrollo e Inteligencia de
Negocios

GRUPO [GESFOR OSMOS](#) PERU S.A.

Microsoft Certified System Engineer

Microsoft Certified Systems Administrator

Microsoft Certified DataBase Administrator

[robertancajima\(at\)](mailto:robertancajima(at)informatizate(dot)net)

[informatizate\(dot\)net](mailto:informatizate(dot)net)

Julio 13 del 2004.



La seguridad es uno de las principales preocupaciones de los administradores de sistemas, redes y base de datos. Mientras un administrador esta implementando la seguridad es natural que pueda estar preocupado por los ataques externos. Pero existe más que eso. Es esencial primero implementar la seguridad dentro de la organización, para asegurar que solo las personas autorizadas tengan acceso a los datos. Si no se establece estas medidas de seguridad, lo mas seguro es encontrar a alguien destruyendo nuestros datos o vendiendo los secretos de la compañía a nuestros competidores, o alguien accediendo a la información privada de otros. Lo primero que se tiene que hacer es un plan de seguridad, el cual identifique que usuarios en la organización pueden ver determinados datos y que actividades deben realizar en la base de datos.

Modelo de Seguridad de SQL Server.

Para poder acceder a los datos de una base de datos, un usuario tiene que pasar a través de 2 niveles de autenticación: uno en el nivel de SQL Server y el otro en el nivel de base de datos. Estos 2 niveles son implementados usando nombres de logias y cuentas de usuario, respectivamente. Un login valido es requerido para conectarse al SQL Server y una cuenta de usuario valido es requerido para acceder a una base de datos.

- **Login:** Un nombre de login válido es necesario para conectarse a una instancia de SQL Server. Un login puede ser:
 - Un login de Windows NT/2000/2003 que ha sido concedido el acceso a SQL Server.
 - Un login de SQL Server, que es manejado dentro del SQL Server.Estos nombres de login son manejados dentro de la base de datos master. Esta es una de las razones del porqué es esencial realizar una copia de seguridad después de adicionar nuevos logins a SQL Server.
- **User:** Una cuenta de usuario válida dentro de una base de datos es requerida para acceder a esta base de datos. Las cuentas de usuario son específicas a una base de datos. Todos los permisos y los propietarios de los objetos son controlados por la cuenta de usuario. Los logins de SQL Server son asociados con estas cuentas de usuarios. Un login puede tener usuarios asociados en diferentes bases de datos, pero únicamente un usuario por base de datos.

Durante una solicitud de nueva conexión, SQL Server verifica el nombre de login suministrado, para asegurar que el login esta autorizado para acceder a SQL Server. Este proceso de verificación es llamado autenticación. SQL Server soporta 2 modos de autenticación:

Modo de Autenticación de Windows: Con la autenticación de Windows, no necesitamos especificar un nombre y contraseña de login para conectarse a SQL Server. Debido a que el acceso a SQL Server es controlado por nuestra cuenta de usuario de Windows NT/2000 (o el grupo al cual nuestra cuenta pertenece), que es usado para acceder al sistema operativo en la computadora cliente. Un DBA tiene que especificar a SQL Server todas las cuentas o grupos de Microsoft Windows NT/2000/2003 que pueden conectarse a SQL Server.

Modo Mixto: Este modo de autenticación permite a los usuarios conectarse usando autenticación de Windows o autenticación de SQL Server. Nuestro DBA primero tiene que crear una cuenta y contraseña de login válida en SQL Server. Estas cuentas no están relacionadas a las cuentas de Microsoft Windows NT/2000/2003. Con este modo de autenticación, tenemos que suministrar el login y contraseña de SQL Server cuando nos conectamos a SQL Server. Si no especificamos un nombre de login y contraseña de SQL Server o solicitamos Autenticación de Windows, nosotros estaremos autenticándonos usando Autenticación de Windows.

Un punto a tener en cuenta es que cualquier modo que se configure en nuestro SQL Server, podemos siempre utilizar la Autenticación de Windows.

La Autenticación de Windows es el modo de seguridad recomendado, debido a que es más seguro y no tenemos que enviar nombres de login y contraseñas sobre la red. Nosotros deberíamos evitar usar el Modo Mixto, al menos que tengamos un ambiente que no sea Windows NT/2000/2003, o cuando nuestro SQL Server sea instalado en un Windows 98, o por compatibilidad con aplicaciones existentes.

El modo de autenticación de SQL Server puede ser cambiado a través de la herramienta Administrador Corporativo (clic derecho en el nombre del servidor y clic en "Propiedades", luego ir al Tab "Seguridad").

El modo de autenticación también puede ser cambiado utilizando el modelo de objeto SQL DMO, permitiendo a los desarrolladores escribir programas para manejar la seguridad de SQL Server.

Aquí esta una lista de procedimientos almacenados útiles para manejar logins y usuarios:

sp_addlogin: Crea un login y permite a los usuarios conectarse a SQL Server usando autenticación SQL Server.

sp_grantlogin: Permite a una cuenta o grupo de usuarios de Windows NT/2000/2003 conectarse a SQL Server usando Autenticación de Windows.

sp_droplogin: Elimina un login de SQL Server.

sp_revokelogin: Elimina un login de Windows NT/2000/2003 de SQL Server.

sp_denylogin: Previene a un login de Windows NT/2000 de conectarse a SQL Server.

sp_password: Adiciona o cambia la contraseña del login de SQL Server.

sp_helplogins: Proporciona información acerca de los logins y sus usuarios asociados en cada base de datos.

sp_defaultdb: Cambia la base de datos por defecto de un login.

sp_grantdbaccess: Adiciona un usuario asociado en la base de datos actual para un login de SQL Server o Windows NT/2000/2003.

sp_revokedbaccess: Elimina una cuenta de usuario de la base de datos actual.

sp_helpuser: Reporta información acerca de los usuarios y roles de Microsoft en la actual base de datos.

Ahora vamos a hablar acerca de cómo controlar el acceso a los objetos dentro de una base de datos y el manejo de los permisos. Aparte de manejar los permisos en el nivel de usuario de base de datos individual, SQL Server 7.0 y 2000 implementa permisos usando roles. En síntesis un rol no es más que un grupo de al cual los logins y usuarios individuales pueden ser adicionados, para que los permisos puedan ser aplicados a un grupo, en lugar de aplicar los permisos a todos los usuarios y logins en forma individual.

Existen 3 tipos de roles en SQL Server 7.0 y 2000:

- Roles de Servidor
- Roles de Base de Datos
- Roles de Aplicación

Roles de Servidor

Son los roles que son aplicados a nivel del servidor de SQL Server. Los logins pueden ser adicionados a estos roles para ganar los permisos administrativos asociados de este rol. Un rol de servidor no puede ser alterado y un nuevo rol de servidor no puede ser creado. Los roles de servidor y los permisos asociados a estos en SQL Server 2000 son:

sysadmin: Puede realizar cualquier actividad en SQL Server.

serveradmin: Puede establecer opciones de configuración a nivel de servidor, como por ejemplo apagar el servidor.

setupadmin: Puede manejar servidores enlazados (linked server) y procedimientos de inicio.

securityadmin: Puede manejar logins y permisos de CREATE DATABASE, también leer los registros de error y cambio de contraseñas.

processadmin: Puede manejar procesos en SQL Server.

dbcreator: Puede crear, alterar o eliminar bases de datos.

diskadmin: Puede manejar archivos de disco.

bulkadmin: Puede ejecutar sentencias BULK INSERT.

Aquí está una lista de procedimientos almacenados que pueden ser muy útiles en el manejo de roles de servidor:

sp_addsrvrolemember: Adiciona un login como un miembro de un rol de servidor.

sp_dropsrvrolemember: Remueve un login de SQL Server, usuario o grupo de Windows de un rol de servidor.

sp_helpsrvrole: Retorna una lista de roles de servidor.

sp_helpsrvrolemember: Retorna información acerca de los miembros de los roles de servidor.

sp_srvrolepermission: Retorna los permisos aplicados a un role de servidor.

Roles de Base de datos

Cada base de datos tiene un conjunto definido de roles de base de datos, al cual los usuarios de base de datos pueden ser adicionados. Estos roles de base de datos son únicos dentro de la base de datos. Mientras los permisos de los roles de base de datos no pueden ser alterados, nuevos roles de base de datos pueden ser creados. Los roles de base de datos y sus permisos asociados en SQL Server 2000 son los siguientes:

dbowner: Tiene todos los permisos en la base de datos.

db_accessadmin: Puede adicionar o remover usuarios.

db_securityadmin: Puede manejar todos los permisos, propiedad de los objetos.

db_ddladmin: Puede usar todas las sentencias del lenguaje de definición de datos, menos GRANT, REVOKE o DENY.

db_backupoperator: Puede usar las sentencias DBCC, CHECKPOINT y BACKUP.

db_datareader: Puede seleccionar todos los datos de cualquier tabla de usuario en la base de datos.

db_datawriter: Puede modificar cualquier dato en cualquier tabla de usuario en la base de datos.

db_denydatareader: No puede seleccionar algún dato de cualquier tabla en la base de datos.

db_denydatawriter: No puede modificar algún dato de cualquier tabla de usuario en la base de datos.

Aquí esta una lista de procedimientos almacenados que pueden ser muy utilies en el manejo de los roles de base de datos.

sp_addrole: Crea un nuevo rol de base de datos en la base de datos actual.

sp_addrolemember: Adiciona un usuario a un rol de base de datos existente en la base de datos actual.

sp_dbfixedrolepermission: Muestra los permisos para cada rol de base de datos.

sp_droprole: Remueve un role de base de datos de la base de datos actual.

sp_helpdbfixedrole: Retorna una lista de roles de base de datos.

sp_helprole: Retorna información acerca de los roles en la base de datos actual.

sp_helprolemember: Retorna información acerca de los miembros de un rol en la base de datos actual.

sp_droprolemember: Remueve usuarios de un rol específico en la base de datos.

Roles de Aplicación

Estos roles son un poco diferente de los roles de servidor y base de datos. Después de crear y asignar los permisos requeridos a un rol de aplicación, la aplicación cliente necesita activar este rol en tiempo de ejecución para obtener los permisos asociados con este rol de aplicación. Los roles de aplicación simplifican el trabajo de los administradores de base de datos, debido a que ellos no estarían preocupados en manejar los permisos a nivel de usuario individual. Todo lo que necesitan hacer es crear un rol de aplicación y asignar permisos a este rol. La aplicación que está conectándose a la base de datos activa el rol de aplicación y hereda los permisos asociados con este rol. Algunas de las características de los roles de aplicación son los siguientes:

- No existen roles de aplicación pre-definidos.
- Los roles de aplicación no contienen miembros.
- Los roles de aplicación necesitan ser activados en tiempo de ejecución, por la aplicación, usando una contraseña.
- Los roles de aplicación sobrescriben los permisos estándar. Por ejemplo, después de activar el rol de aplicación, la aplicación perderá los permisos asociados con el login o la cuenta de usuario usado mientras se conecta a SQL Server y obtiene los permisos asociados con el rol de aplicación.
- Los roles de aplicación son específicos a la base de datos. Después de activar un rol de aplicación en una base de datos, si la aplicación quiere ejecutar una transacción a través de base de datos, la otra base de datos tienen que tener habilitado una cuenta de usuario invitado.

Aquí están los procedimientos almacenados que son requeridos para manejar los roles de aplicación:

sp_addrole: Adiciona un rol de aplicación en la base de datos actual.

sp_approlepassword: Cambia la contraseña de un rol de aplicación en la base de datos actual.

sp_dropapprole: Elimina un rol de aplicación de la base de datos actual.

sp_setapprole: Activa los permisos asociados con un rol de aplicación en la base de datos actual.

Permisos

Luego de estudiar los diferentes tipos de roles, vamos a conocer acerca de los permisos de concesión y revocación hacia y desde los usuarios de base de datos y los roles de base de datos y roles de aplicación. Los siguientes comandos de T-SQL son usados para manejar los permisos en el nivel de usuario y rol:

GRANT: Concede los permisos específicos (SELECT, DELETE, etc.) a un usuario o rol específico en la base de datos actual.

REVOKE: Remueve un permiso previamente establecido, ya sea GRANT o DENIED de un usuario o rol en la base de datos actual.

DENY: Niega un permiso específico a un usuario o rol específico de la base de datos actual.

Usando los comandos mencionados anteriormente, los permisos pueden ser concedidos, denegados o revocados a usuarios y roles en todos los objetos de la base de datos. Nosotros podemos también manejar permisos a nivel de columna.

En SQL server no existe forma para manejar permisos en el nivel de fila. Esto es en una determinada tabla, no podemos conceder el permiso de SELECT en una fila específica a un Usuario1 y denegar el permiso SELECT en otra fila al Usuario2. Esta forma de seguridad puede ser implementada creando vistas específicas para usuarios y conceder el permiso SELECT en estas vistas a los usuarios. Pero esta solución puede llegar a ser un poco tediosa en el caso de tener muchos usuarios con requerimientos de datos variados.

La seguridad es un punto crítico por el cual es muy importante que todo administrador de base de datos tenga los conceptos claros del modelo de seguridad de SQL Server 7.0/2000 con el objetivo de establecer un plan de seguridad efectivo en su organización.

REFERENCIAS:

Administering Database Security, Microsoft SQL Server 2000 Books On-line.

Course 2072: Administering a Microsoft SQL Server 2000 Database, Microsoft Official Curriculum.

Microsoft SQL Server 2000 Database Design and Implementation, Microsoft Press.