Seguridad en SQL Server

Microsoft

Jesús López

- MVP Visual Basic – Visual Developer
- Formador en Alhambra-Eidos
- Mentor Asociado Solid Quality Learning
- Consultor y desarrollador

Microsoft

Solid Quality Learning

Una asociación de expertos en SQL Server por todo el mundo

- Mentores principales:
  - Itzik Ben-Gan
  - Kalen Delaney
  - Fernando G. Guerrero
  - Michael Hotek
  - Brian Moran
  - Ron Talmage
- Mentores asociados:
  - Herbert Albert
  - Gianluca Hotz
  - Tibor Karaszi
  - Andrew Kelly
  - Dejan Sarka
  - Wayne Snyder
  - Jesús López
  - Eladio Rincón
  - Miguel Egea

Te ayudan a sacarle el mayor partido a SQL Server. Formación y Consultoría

Microsoft

Agenda

- Arquitectura de Seguridad en SQL Server 2000
- Creación y Gestión de inicios de sesión
- Gestión de permisos
- Estrategias de seguridad
- Novedades de Seguridad en SQL Server 2005

Microsoft

Arquitectura de seguridad en SQL Server 2000

Microsoft

Aspectos de la seguridad

- Autentificación:
  - Verificar la identidad del usuario
- Autorización:
  - Permitir o no acceder a un recurso o realizar una determinada acción en función de la identidad del usuario
- Integridad:
  - Asegurar que la información no ha sido modificada en tránsito. Firma digital
- Privacidad:
  - Asegurar que la información sólo puede ser leída por el destinatario. Cifrado

Microsoft

Tipos de autentificación

- Windows (integrada o de confianza)
  - SQL Server delega en Windows para realizar la autentificación.
  - Sujetas a las directivas de Windows
- SQL Server (o estándar)
  - Nombre de usuario y contraseña (hash) guardada en SQL Server. Requeridos en cada establecimiento de conexión.

Configuración de la autentificación

- SQL Server permite dos modos:
  - Windows (predeterminado)
  - SQL Server y Windows
- Establecido en la instalación
- Puede cambiarse posteriormente:
  - Administrador corporativo
  - Valor LoginMode en el registro
  - Requiere reinicio del servicio
- Saber el modo actual:
  - SERVERPROPERTY('IsIntegratedSecurityOnly')

Delegación: Kerberos

- Permite que una instancia de SQL Server se conecte con otra instancia en otra máquina con las credenciales del cliente.
- Es necesario:
  - Directorio activo
  - Usar protocolo TCP-IP
  - Confiar en el equipo para realizar delegación
  - Un nombre principal de servicio para SQL Server (setspn)
  - Las cuentas del cliente tiene que ser delegables.
- Ver artículo 319723 : “INF: SQL Server 2000 Kerberos support including SQL Server virtual servers on server clusters”

Microsoft

Autorización: gestión de permisos

- Cuenta de inicio de sesión requerido para establecer conexión
- La autorización puede establecerse por medio de:
  - Roles de servidor
  - Roles de bases de datos
  - Permiso de acceso a base de datos
  - Permiso de ejecución de sentencias
  - Permisos específicos para objetos de base de datos

Integridad y privacidad

- Dos mecanismos disponibles en SQL Server:
  - Cifrado multiprotocolo:
    - Debe usarse multiprotocolo tanto en el cliente como en el servidor
    - Sólo funciona con la instancia predeterminada
    - Requiere configuración en el cliente y en el servidor
    - Ver artículo 841695: “How to establish and enforce encrypted multiprotocol connections in SQL Server 2000”
  - SSL:
    - Requiere certificado digital en el Servidor.
    - En el cliente: certificado de la entidad raíz que emitió el certificado del servidor
    - Ver artículo 276553 : “HOW TO: Enable SSL Encryption for SQL Server 2000 with Certificate Server”

Microsoft

Demo

- Configurar la autentificación
  - Administrador corporativo
  - Registro: LoginMode
- Ver la configuración:
  - SELECT SERVERPROPERTY('IsIntegratedSecurityOnly')

Microsoft

Creación y gestión de inicios de sesión

Microsoft

Inicios de sesión

- Permiten o deniegan la conexión a SQL Server
- Definidos a nivel de servidor
- Necesario para conectarse a SQL Server
- Dos tipos basados en la autentificación:
  - SQL Server: nombre + contraseña
  - Windows: usuario o grupo de Windows

Inicios de sesión predefinidos

- Al instalarse SQL Server se crean dos inicios de sesión
- Pueden realizar cualquier tarea en SQL Server (pertenecen al rol de servidor sysadmin)
- BUILTIN\Administradores. (grupo predefinido de Windows). Puede eliminarse y modificarse los permisos.
- sa: tipo estándar. No puede eliminarse ni modificarse. No estará disponible si sólo está configurada la autentificación Windows.

Microsoft

Creación de inicios de sesión

- Sólo sysadmin y securityadmin
- Mediante el Administrador Corporativo
- Mediante procedimientos almacenados
  - SQL Server: sp_addlogin
  - Windows: sp_grantlogin, sp_denylogin
- Información de inicios de sesión:
  - sp_helplogins
  - syslogins

Gestión de inicios de sesión

- Eliminar un inicio de sesión:
  - Sólo sysadmin y securityadmin
  - sp_droplogin (SQL Server)
  - sp_revokelogin (Windows)
  - No se puede si:
    - Es propietario de alguna base de datos
    - Es usuario de alguna base de datos
    - Es propietario de algún objeto de alguna base de datos.
- Cambiar contraseña (sólo SQL Server)
  - sp_password
  - Sólo sysadmin, securityadmin y el propio usuario.
- Propiedades:
  - Base de datos predeterminada: sp_defaultdb
  - Lenguaje predeterminado: sp_defaultlanguage
  - Sólo sysadmin, securityadmin y el propio usuario.

Microsoft

Demo

- Creación y gestión de inicios de sesión:
  - Administrador Corporativo:
    - Crear inicio sesión SQL Server y Windows
    - Establecer base de datos y lenguaje predeterminados
    - Eliminar inicio de sesión
  - T-SQL:
    - Crear y gestionar inicios de sesion.sql

Gestión de permisos

Microsoft

Roles de servidor

- Cada rol agrupa un conjunto de permisos.
- Facilitan la administración de la seguridad.
- Definidos a nivel de servidor. Independientes de las bases de datos.
- Un inicio de sesión puede pertenecer a cero o más roles de servidor
- Un inicio de sesión que pertenezca a un rol de servidor adquiere los permisos de ese rol.
- Son fijos:
  - No pueden modificarse sus permisos
  - No pueden eliminarse
  - No pueden añadirse nuevos roles de servidor

Microsoft

Lista de roles de servidor

- sysadmin: puede realizar cualquier actividad en SQL Server
- securityadmin: administra los inicios de sesión
- serveradmin: configura el servidor
- setupadmin: configura servidores vinculados y procedimiento almacenado de arranque
- processadmin: puede desconectar usuarios
- diskadmin: (obsoleto 6.5)
- dbcreator: crea bases de datos
- bulkadmin: realiza inserciones masivas

Gestión de roles de servidor

- Obtener información (todos pueden):
  - sp_helpsrvrole. Lista de roles
  - sp_helpsrvrolemember. Lista de miembros
  - syslogins: una columna por rol de servidor
- Añadir un inicio de sesión a un rol de servidor:
  - Sólo sysadmin y miembros del rol
  - sp_addsrvrolemember
- Quitar un inicio de sesión de un rol de servidor:
  - Sólo sysadmin y miembros del rol
  - sp_dropsrvrolemember

Demo

- Gestión de roles de servidor
  - Administrador Corporativo:
    - Lista de roles de servidor
    - Permisos de los roles de servidor
    - Miembros de un rol: ver, añadir y quitar.
    - Roles a los que pertenece un inicio de sesión: ver, añadir y quitar.
  - T-SQL:
    - Roles de servidor.sql

Acceso a una base de datos

- Los siguientes pueden conectarse:
  - sysadmin
  - propietario de la base de datos
  - Los usuarios de la base de datos
  - Cualquier inicio de sesión si existe el usuario guest.
- Usuario de base de datos:
  - Definido a nivel de base de datos
  - Corresponde con un inicio de sesión
  - dbo: usuario predefinido. (data base owner). Corresponde con sysadmin, propietario de base de la base de datos, rol db_owner.
  - guest: invitado. Puede existir o no

Conceder acceso a una base de datos

- Pueden:
  - sysadmin
  - Propietario
  - db_owner
  - db_accessadmin
- Al conceder acceso a un inicio de sesión a una base de datos:
  - se crea el usuario correspondiente en esa base de datos
  - Pertenece al rol public
- Conceder acceso: sp_grantdbaccess y sp_adduser
- Quitar el acceso: sp_revokedbaccess
- Reasignar Inicios de sesión con usuarios. Útil cuando se anexa o restaura una base de datos: sp_change_users_login
- Info: sp_helpuser

Propietario de una base de datos

- Puede realizar cualquier operación en la base de datos.
- Es en principio quien crea la base de datos
- Puede cambiarse con sp_changedbowner
- Saber el propietario: sp_helpdb

Demo

- Propietario de una base de datos y gestión de usuarios
  - Administrador Corporativo:
    - Ver propietario de base de datos
    - Ver usuarios de base de datos
    - Crear usuario
    - Eliminar usuario
  - T-SQL:
    - Acceso y propietario base datos.sql

Roles de bases de datos

- Cada rol agrupa un conjunto de permisos.
- Facilitan la administración de la seguridad.
- Definidos a nivel de base de datos.
- Un usuario puede pertenecer a uno o más roles de base de datos
- Un usuario que pertenezca a un rol de base de datos adquiere los permisos de ese rol.
- Un rol puede contener otros roles.
- Tres tipos:
  - Predefinidos
  - Definidos por el usuario:
    - Estándar
    - De aplicación

Microsoft

Roles predefinidos de base de datos

- Son fijos:
  - No pueden modificarse sus permisos. Excepto el de Public
  - No pueden eliminarse
  - No pueden añadirse nuevos roles predefinidos
- Los siguientes:
  - db_owner: realizan cualquier operación en la base de datos
  - db_backupoperator: realiza copias de seguridad
  - db_datareader: lee de cualquier tabla
  - db_datawriter: escribe en cualquier tabla
  - db_denydatawriter: no puede escribir en ninguna tabla
  - db_denydatareader: no puede leer de ninguna tabla
  - db_ddladmin: crea y modifica objetos en la base de datos
  - db_accessadmin: gestiona usuarios en la base de datos
  - db_securityadmin: gestiona los permisos

Información de roles predefinidos de base de datos

- Lista: sp_helpdbfixedrole (todos)
- Ver los permisos: sp_dbfixedrolepermission (todos)

Roles definidos por el usuario

- Agrupan un conjunto de permisos
- No tienen permisos predefinidos
- Los permisos se establecen por:
  - Pertenencia a otros roles (predefinidos y estándar definidos por el usuario)
  - Permisos de sentencias
  - Permisos específicos de objetos.
- Dos tipos:
  - Estándar
  - De aplicación
- Pueden gestionarlos: sysadmin, propietario, db_owner y db_securityadmin

Roles estándar definidos por el usuario

- Pueden contener usuarios y roles definidos por el usuario
- Pueden pertenecer a roles predefinidos y roles estándar
- Lista: sp_helprole
- Añadir: sp_addrole
- Eliminar: sp_droprole

Roles de aplicación

- Pueden pertenecer a roles predefinidos y roles estándar.
- No pueden contener miembros
- Tienen asociada una contraseña
- Son activados por una aplicación cliente por medio de sp_setapprole.
- Una vez activado la conexión adquiere los derechos de rol, pierde los que tenía, y ya no puede acceder a ninguna otra base de datos.
- Crear: sp_addapprole
- Eliminar: sp_dropapprole
- Cambiar contraseña: sp_approlepassword

Microsoft

Gestión de roles: pertenencia

- Pueden añadir y quitar: sysadmin, propietario, db_owner y db_securityadmin
- Añadir un miembro al rol (puede añadirse un usuario o un rol definido por el usuario): **sp_addrolemember**
- Quitar un miembro del rol: **sp_droprolemember**
- Ver los roles a los que pertenece un usuario: **sp_helpuser**
- Ver los roles y sus miembros:

```
SELECT    roles.name As Rol, members.name As Miembro
FROM      sysusers roles
          JOIN sysmembers mr
          ON roles.uid = mr.groupuid
          JOIN sysusers members
          ON members.uid = mr.memberuid
```

Microsoft

Demo

- Roles de bases de datos:
  - Administrador corporativo:
    - Ver roles predeterminados
    - Añadir y quitar miembros al rol
    - Añadir y quitar roles definidos por el usuario
  - T-SQL:
    - Roles de base de datos.sql

Permisos de sentencias

- A un usuario, rol definido por el usuario, y a public, puede concederse o denegarse permiso para ejecutar sentencias
- GRANT <Sentencia> TO <Usuario o Rol>
- DENY <Sentencia> TO <Usuario o Rol>
- REVOKE <Sentencia> FROM <Usuario o Rol>
- Obtener información: sp_helprotect

Lista de sentencias

- CREATE DATABASE
- CREATE DEFAULT
- CREATE FUNCTION
- CREATE PROCEDURE
- CREATE RULE
- CREATE TABLE
- CREATE VIEW
- BACKUP DATABASE
- BACKUP LOG

Demo

- Permisos de sentencias
  - Administrador corporativo:
    - Ver y modificar permisos de sentencias
  - T-SQL:
    - Permisos de sentencias.sql

Microsoft

Permisos específicos de objetos

- A un usuario, rol definido por el usuario, y a public, se les puede conceder o denegar permisos sobre objetos
- GRANT <Permiso> ON <Objeto> TO <Usu>
- DENY <Permiso> ON <Objeto> TO <Usu>
- REVOKE <Permiso> ON <Objeto> FROM <Usu>
- Obtener información: sp_helprotect

Objetos y permisos

Objeto/permiso	Select	Insert	Update	Delete	Execute	References
Tabla / Vista / función en línea	x	x	x	x		x
Función tabular	x					x
Procedimiento almacenado					x	
Función escalar					x	x

Permisos efectivos

- PE = C – D
- $C = O_c + S_c + R_c$
- $D = O_d + S_d + R_d$

- PE: permisos efectivos
- C: permisos concedidos
- D: permisos denegados
- O_c: permisos de objeto concedidos
- S_c: permisos de sentencias concedidos
- R_c: permisos concedidos de los roles a los que pertenezca
- O_d: permisos de objeto denegados
- S_d: permisos de sentencias denegados
- R_d: permisos denegados de los roles a los que pertenezca

Demo

- Permisos de objetos:
  - Administrador corporativo:
    - Ver y modificar permisos de objetos
  - T-SQL:
    - Permisos de objeto.sql

Microsoft

Estrategias de seguridad

Microsoft

Uso de vistas y funciones en línea

- Dar permisos a vistas y funciones en línea en lugar de a las propias tablas
- Ocultan la complejidad de la base de datos
- Permiten fácilmente gestionar el acceso a nivel de columna.

Uso de procedimientos almacenados

- Algunos administradores sólo permiten el acceso por medio de procedimientos almacenados porque:
  - Impiden operaciones incorrectas asegurando las reglas de negocio.
  - Permiten afinar la seguridad al mayor grado.
- Los usuarios no necesitan tener permiso para acceder a las tablas, sólo permiso de ejecución de los procedimientos almacenados.
- Cuando un usuario A ejecuta un procedimiento almacenado B.P SQL Server no comprueba los permisos de acceso de A al los objetos de B. Sólo comprueba los permisos de acceso de los objetos que no son de B. Esto es cierto sólo para las sentencias DML, para las sentencias DDL siembre se comprueban los permisos.

Formas de controlar la seguridad

- Cuando se diseña un sistema se tienen básicamente dos opciones para controlar la seguridad:
  - Basarse en la seguridad de SQL Server.
    - Ventajas: facilidad de auditoría
    - Inconvenientes: los usuarios pueden acceder a la base de datos por otros medios y hacer de las suyas.
  - La aplicación cliente controla la seguridad:
    - Ventajas: los usuarios sólo pueden acceder a la base de datos por medio de la aplicación
    - Inconvenientes: dificultad de auditoría

Seguridad basada en SQL Server

- Elegir entre dos tipos de autentificación:
  - Windows (Recomendada)
  - SQL Server

Microsoft

Configurar la seguridad basada en SQL Server con autentificación Windows

- Crear un grupo de Windows por cada perfil de usuario
- Añadir los usuarios de Windows al grupo correspondiente
- Crear un inicio de sesión en SQL Server por cada grupo de Windows correspondiente a un perfil
- Conceder los permisos necesarios a cada grupo de Windows.

Microsoft

Utilizar la seguridad basada en SQL Server con autentificación Windows

- En una aplicación ASP.NET
  - Utilizar autentificación Windows
  - Configurarla para suplantación
  - Desventaja: no connection pooling, no escalable
- En una aplicación COM+ de servidor:
  - Utilizar suplantación
  - Desventaja: no connection pooling, no escalable
- En una aplicación Windows:
  - Nada

Microsoft

Configurar la seguridad basada en SQL Server con autentificación estándar

- Crear un rol estándar por cada perfil de usuario
- Conceder los permisos necesarios a cada rol
- Crear un inicio de sesión por cada usuario de la aplicación y:
  - Concederle acceso a la base de datos
  - Añadirle al rol correspondiente a su perfil.

Microsoft

Utilizar la seguridad basada en SQL Server con autentificación estándar

- En una aplicación ASP.NET
  - Autentificación Forms
  - Desventaja: no connection pooling
- En una aplicación COM+ de servidor
  - No sería aplicable
- En una aplicación Windows:
  - Formulario de inicio de sesión.

Microsoft

Seguridad basada en aplicación

- Autentificación:
  - Manejada por la propia aplicación
    - Windows. Ej (ASP.NET, WinApp, COM+)
    - Personalizado. Ej. (ASP.NET, WinApp)
  - Basada en SQL Server (Windows o estándar)
- Conexión a SQL Server
  - Qué credenciales y qué tipo de autentificación
- Usuario y permisos:
  - Un inicio de sesión estándar
  - Un inicio de sesión usuario de windows
  - Un rol de aplicación

Microsoft

Seguridad basada en la aplicación: ejemplos

- ASP.NET
  - Autentificación Forms y seguridad basada en roles.
  - Utiliza un usuario estándar específico de la aplicación que es db_ddladmin, db_datareader y db_datawriter en la base de datos.
- COM+:
  - Seguridad basada en roles
  - Conexión: autentificación Windows
  - Usuario: Identidad de la aplicación. Es db_datareader y db_datawriter en la base de datos.
- Aplicación Windows:
  - Hay un grupo de Windows por cada perfil de usuario
  - Cada grupo tiene un inicio de sesión en SQL Server, pero no tiene acceso a ninguna base de datos.
  - Utiliza autentificación Windows para conectarse con SQL Server
  - Activa un rol de aplicación justo después de la conexión que tiene permiso de ejecución de todos los procedimientos almacenados.
  - Utiliza seguridad basada en roles

Microsoft

Seguridad en SQL Server 2005

Microsoft

Separación de esquema-usuario

- Separación de principales y esquemas
  - Principal
    - Cuenta de seguridad
    - Se encuentran en la vista sys.database_principals
  - Esquema
    - Contenedor de objetos
    - Se encuentran en sys.schemas
- Noción de esquema predeterminado
  - Es una propiedad del usuario o rol de aplicación
  - Utilizado en la resolución de nombres
  - Están en sys.database_principals
- Nuevas sentencias DDL para usuarios y esquemas
  - CREATE/ALTER/DROP for USER, ROLE and SCHEMA
- La eliminación de usuarios no requerie reescribir las aplicaciones

Microsoft

Demo

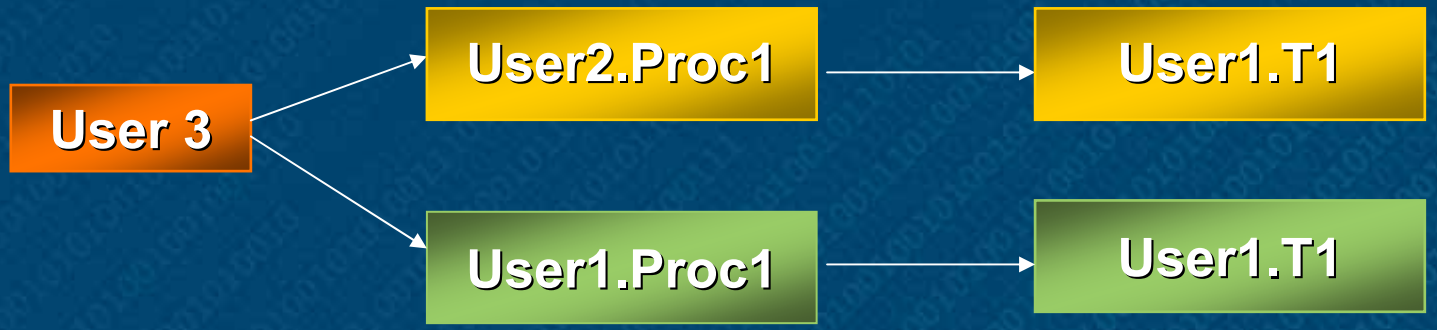
- Esquemas
  - Schemas.sql

Contexto de ejecución

SQL Server 2000

Se comprueba que User3 tiene permiso de ejecución

Se comprueban que User3 tiene permiso Select



SQL Server 2005

Se comprueba que User3 tiene permiso de ejecución

NO se comprueba que User3 tenga permiso

'Execute AS 'X' '

Se comprueba que User3 tiene permiso de ejecución

Permiso select comprobado para 'X'. No para user3 (si el esquema User2 tiene el mismo propietario que el esquema user1 entonces no se comprueba)

Microsoft

Contexto de ejecución de módulos

- Declarara contexto de ejecución de módulos
  - Módulos: procedimientos, funciones, triggers
  - No se necesita delegar en el encadenamiento de la propiedad para que sólo se compruebe el permiso de ejecución
  - Las reglas de encadenamiento de la propiedad se siguen aplicando
- Los permisos se comprueban para el contexto de ejecución actual
  - A diferencia del encadenamiento de propiedad, se aplica también a las sentencias DDL
- Incluye ejecución dinámica
- Se pueden ver en sys.sql_modules

Microsoft

Contexto de ejecución

- Execute AS CALLER
  - Las sentencias se ejecutan con el contexto de, llamador immediato
  - Funcionamiento predeterminado, similar a SQL Server 2000
- Execute AS '*NombreUsuario*'
  - Las sentencias se ejecutan como el usuario especificado
  - Se requiere permiso de suplantación
- Execute AS SELF
  - Las sentencias se ejecutan como el usuario que creó el módulo
  - Se almacena el SID, no 'SELF'
- Execute AS OWNER
  - Las sentencias se ejecutan como el propietario del módulo
  - Se requiere privilegio de suplantación de propietario
  - Cuando cambia el propietario también cambia el contexto

Microsoft

Demo

- Contexto de ejecución
  - Contexto ejecución.sql

Microsoft

Control de permisos granular

- Más permisos – a varios niveles
  - Server, Database, Schema, Object, Principal
- No se necesita pertenecer a un rol para realizar tareas
- Vistas del sistema
  - Permisos de base de datos: sys.database_permissions
  - Permisos de servidor: sys.server_permissions view

Microsoft

Esquema general de permisos

- La mayoría de los objetos asegurables tienen los siguientes permisos
  - CONTROL: como los permisos del propietario
  - ALTER: permite cambiar las propiedades del objeto. Tambien permite CREATE/DROP/ALTER en los objetos contenidos
  - ALTER ANY ‘X’: permite modificar cuanquier objeto de topo X
  - Take Ownership: concede el permiso de obtener la propiedad de un objeto

Directivas de contraseñas

- Durante el inicio de sesión:
  - Fuerza
  - Expiración
  - Bloqueo de cuentas
- Sigue la directiva local de contraseñas
- Implementación
  - Nueva API en Windows Server 2003
  - Llamada durante la autentificación.
- En Win2K Server
  - La función de API no está disponible
  - Sólo se soporta la complejidad de contraseñas nativa de SQL server

Seguridad del catálogo

- Las tablas del sistema están implementadas como vistas
- Los metadatos son seguros de forma predeterminada
  - Permisos mínimos para public
  - Seguridad a nivel de fila en las vistas del catálogo
  - Se necesita tener algún permiso sobre un objeto para verlo en una vista de catálogo
  - SA puede verlo todo en el servidor
  - DBO puede verlo todo en la base de datos
- Nuevo permiso para ver los metadatos
  - VIEW DEFINITON
  - Aplicable a nivel de objeto, esquema, base de datos y server

Microsoft

Principios de diseño de la seguridad

- Seguro por diseño
  - Código robusto y seguro
  - Pruebas y análisis
- Seguro de forma predeterminada
  - La configuración predeterminada es un sistama seguro
  - Minimiza la superficie de ataque
    - Instalar solo los componentes necesarios
- Seguro en producción
  - Principio de mínimos privilegios
    - Concecer los permisos mínimos necesarios para trabajar
    - Cuentas de servicio con pocos privilegios
  - Automatización / Asistencia de mantenimiento de software
  - Buenas herramientas para el análisis y la administración de la seguridad

Microsoft

Demo

- Granularidad y seguridad de catálogo
  - Granularidad y seguridad de catálogo.sql

Microsoft

Microsoft