

HARDENING EN SQL SERVER

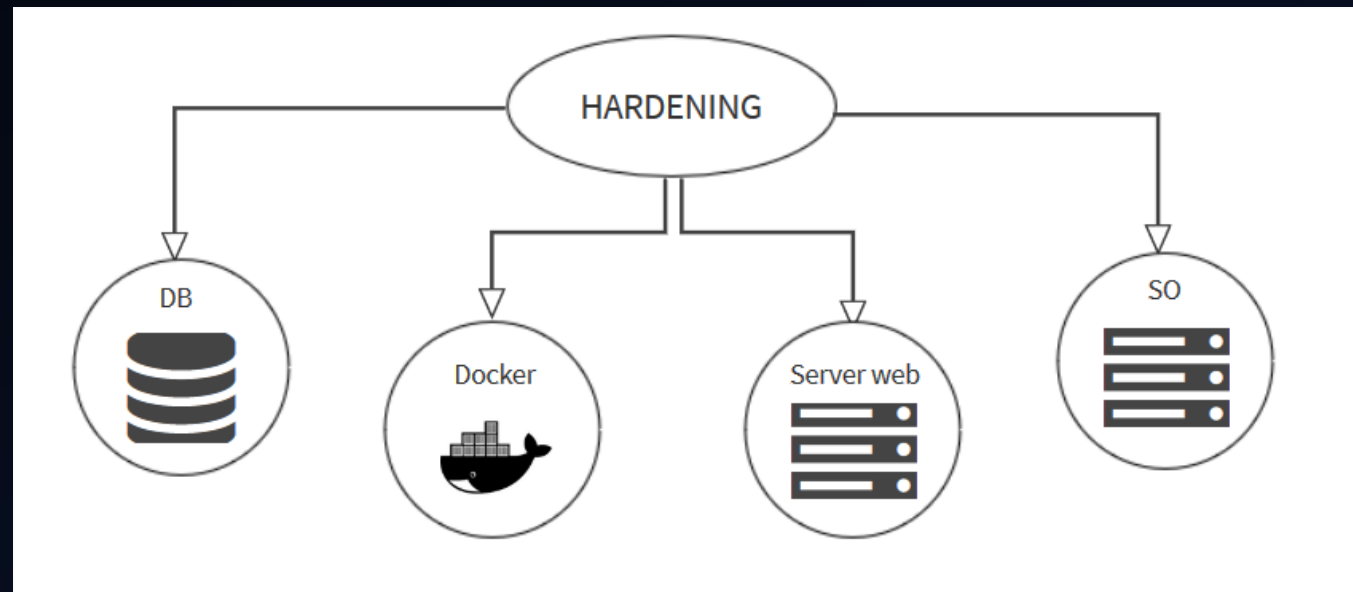
CONFIGURACIÓN DE BASES DE DATOS SEGURAS Y LA IMPORTANCIA DEL ENDURECIMIENTO DE SQL SERVER A TRAVÉS DE LA REDUCCIÓN DE VULNERABILIDADES.



INTRODUCCIÓN

- ¿Qué es hardening?

Es un proceso en el que se realizan configuraciones que permiten asegurar un sistema o servicio mediante la reducción de vulnerabilidades.



¿PORQUÉ DEBERÍA HACER HARDENING EN SQL?

- Para reforzar al máximo la seguridad de mis bases de datos.
- Para proteger la integridad de la información.
- Para entorpecer la labor de un atacante.
- Para ganar tiempo y así minimizar las consecuencias de un incidente de seguridad.

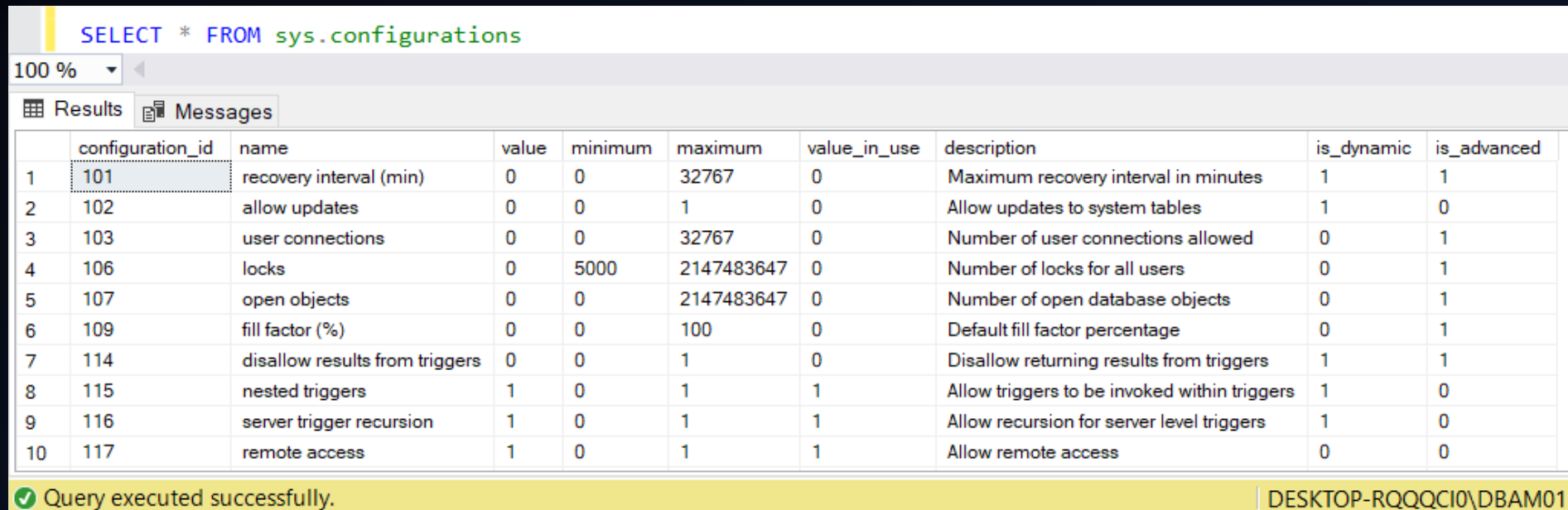
Hacer hardening en nuestras bases de datos no necesariamente las vuelve impenetrables, sin embargo, al convertir esta práctica en una cultura diaria seguro le volverá la labor más difícil a un atacante.

BUENAS PRÁCTICAS DE SEGURIDAD EN SQL SERVER

- Endurezca el servidor que aloja a SQL Server.
- Instale los servicios que son necesarios para su entorno.
- Haga uso de roles y no permisos a usuarios específicos, esto para simplificar la administración de permisos.
- Haga uso de las opciones de control de password (en caso sea posible)
 - ✓ MUST_CHANGE, CHECK_POLICY, CHECK_EXPIRATION
- Instalar actualizaciones de SQL Server.
- Maneje el principio de “menor privilegio”, no todos deben ser sysadmin.
- Cifre la información en caso sea posible.
- Monitoree la actividad en la instancia SQL Server
 - ✓ EVENTOS, CONEXIONES, LOCKS, etc.

SYS.CONFIGURATIONS

- Es un catálogo interno de SQL Server que permite administrar la configuración de la instancia de base de datos.
- En dicho catalogo se almacena un listado de opciones de configuración y estos pueden ser modificados a través del procedimiento “sp_configure”.

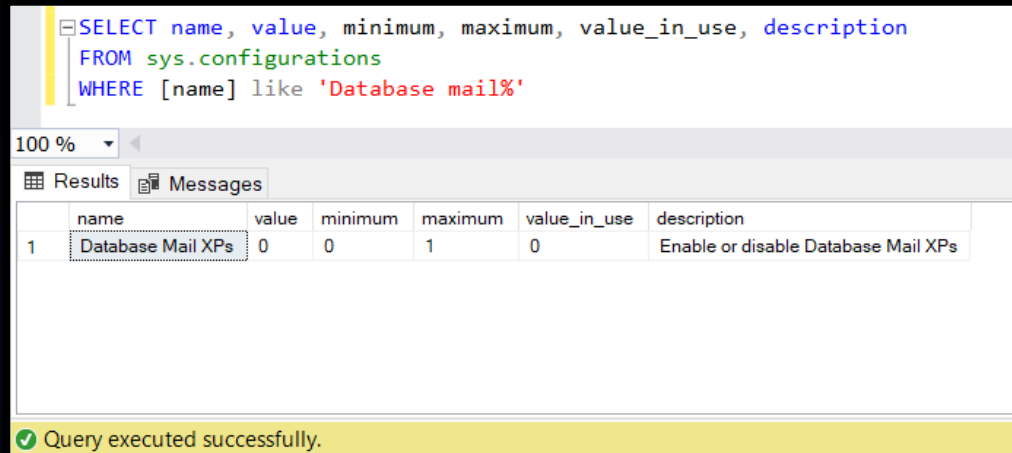


The screenshot displays a SQL Server query window with the following components:

- Query Text:** `SELECT * FROM sys.configurations`
- Execution Status:** 100 %
- Grids:** Results (active) and Messages.
- Results Grid:** A table with 10 rows and 10 columns. The first row is highlighted.
- Status Bar:** Shows a green checkmark and the text "Query executed successfully." followed by the server name "DESKTOP-RQQQCI0\DBAM01".

	configuration_id	name	value	minimum	maximum	value_in_use	description	is_dynamic	is_advanced
1	101	recovery interval (min)	0	0	32767	0	Maximum recovery interval in minutes	1	1
2	102	allow updates	0	0	1	0	Allow updates to system tables	1	0
3	103	user connections	0	0	32767	0	Number of user connections allowed	0	1
4	106	locks	0	5000	2147483647	0	Number of locks for all users	0	1
5	107	open objects	0	0	2147483647	0	Number of open database objects	0	1
6	109	fill factor (%)	0	0	100	0	Default fill factor percentage	0	1
7	114	disallow results from triggers	0	0	1	0	Disallow returning results from triggers	1	1
8	115	nested triggers	1	0	1	1	Allow triggers to be invoked within triggers	1	0
9	116	server trigger recursion	1	0	1	1	Allow recursion for server level triggers	1	0
10	117	remote access	1	0	1	1	Allow remote access	0	0

DATABASE MAIL XPS



The screenshot shows a SQL Server Enterprise Manager interface. At the top, a query window displays the following SQL code:

```
SELECT name, value, minimum, maximum, value_in_use, description
FROM sys.configurations
WHERE [name] like 'Database mail%'
```

Below the query window, the 'Results' tab is active, showing a table with the following data:

	name	value	minimum	maximum	value_in_use	description
1	Database Mail XPs	0	0	1	0	Enable or disable Database Mail XPs

At the bottom of the interface, a yellow status bar indicates: 'Query executed successfully.'

Descripción:

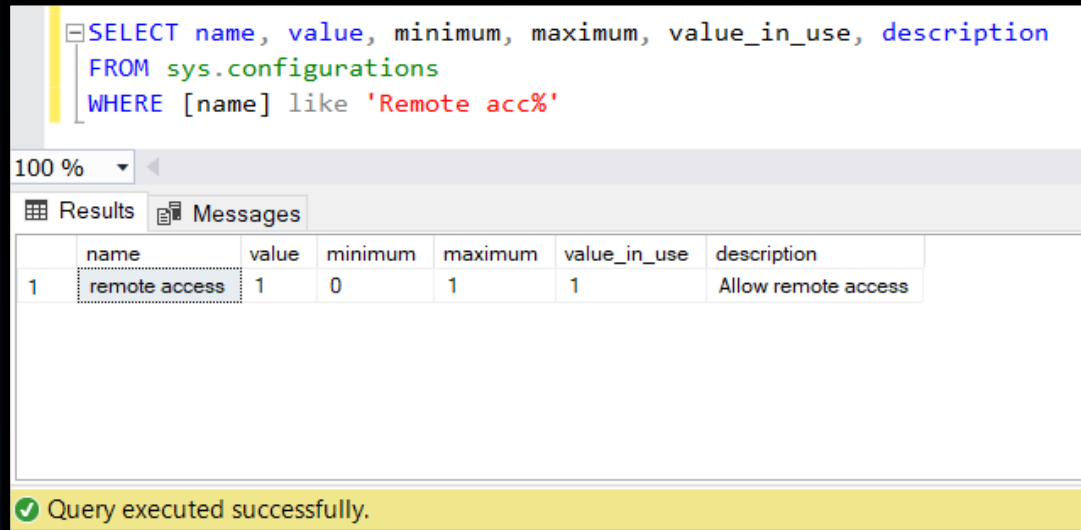
Es una opción que controla la capacidad de generar y transmitir mensajes de correo electrónico desde SQL Server.

Justificación:

Elimina un vector de ataque DOS, con el paso del tiempo se han conocido ataques de este tipo a entidades de gobierno, Facebook, Twitter, etc.

Con este tipo de ataque se busca saturar los servidores de bases de datos hasta que el servicio colapsa y esto provoca que nadie pueda conectarse.

REMOTE ACCESS



```
SELECT name, value, minimum, maximum, value_in_use, description
FROM sys.configurations
WHERE [name] like 'Remote acc%'
```

	name	value	minimum	maximum	value_in_use	description
1	remote access	1	0	1	1	Allow remote access

Query executed successfully.

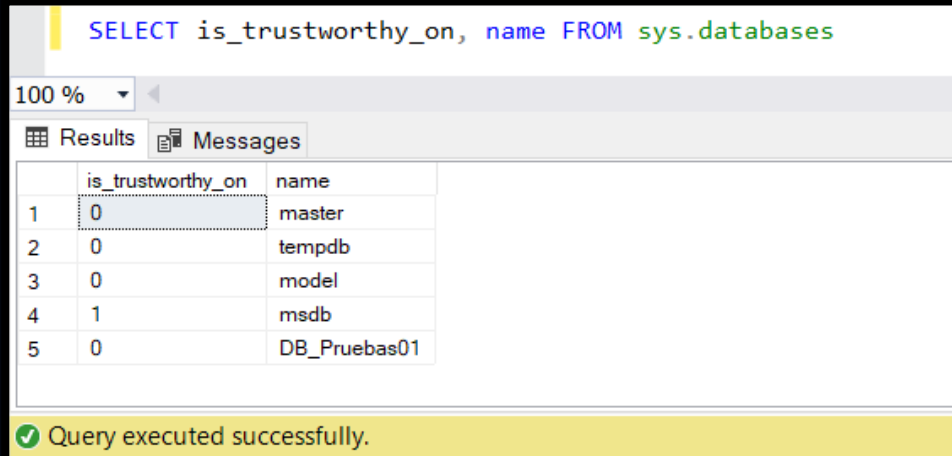
Descripción:

Controla la ejecución de SP's locales en servidores remotos o SP's remotos en servidores locales.

Justificación:

Con esta configuración enabled se puede abusar de la funcionalidad para lanzar un ataque de denegación de servicio DOS en servidores remotos.

TRUSTWORTHY



The screenshot shows a SQL query executed in SQL Server Enterprise Manager. The query is `SELECT is_trustworthy_on, name FROM sys.databases`. The results are displayed in a table with two columns: `is_trustworthy_on` and `name`. The table lists five databases: `master`, `tempdb`, `model`, `msdb`, and `DB_Pruebas01`. The `is_trustworthy_on` values are 0 for `master`, `tempdb`, `model`, and `DB_Pruebas01`, and 1 for `msdb`. A status bar at the bottom indicates "Query executed successfully."

	is_trustworthy_on	name
1	0	master
2	0	tempdb
3	0	model
4	1	msdb
5	0	DB_Pruebas01

Descripción:

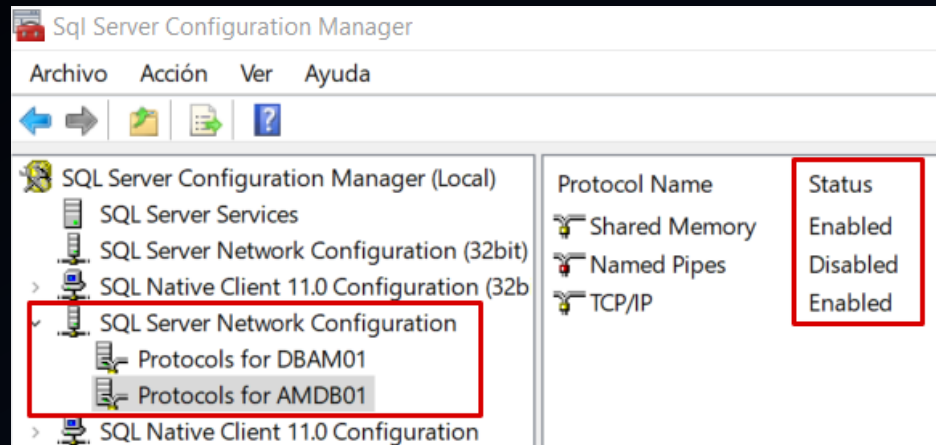
Es una opción a nivel de cada base de datos con la que se puede indicar a la instancia que confíe en una DB y los objetos que se encuentran dentro de ella, es decir, que permite a los usuarios acceder a los objetos de otras bases de datos en ciertas circunstancias, por ejemplo, al hacer un EXECUTE de un SP.

Justificación:

Protege contra ensamblados maliciosos “que tienen configuración” UNSAFE o EXTERNAL ACCESS.

Permite que los privilegios sean controlados de forma mas granular y detallada.

SQL SERVER PROTOCOLS



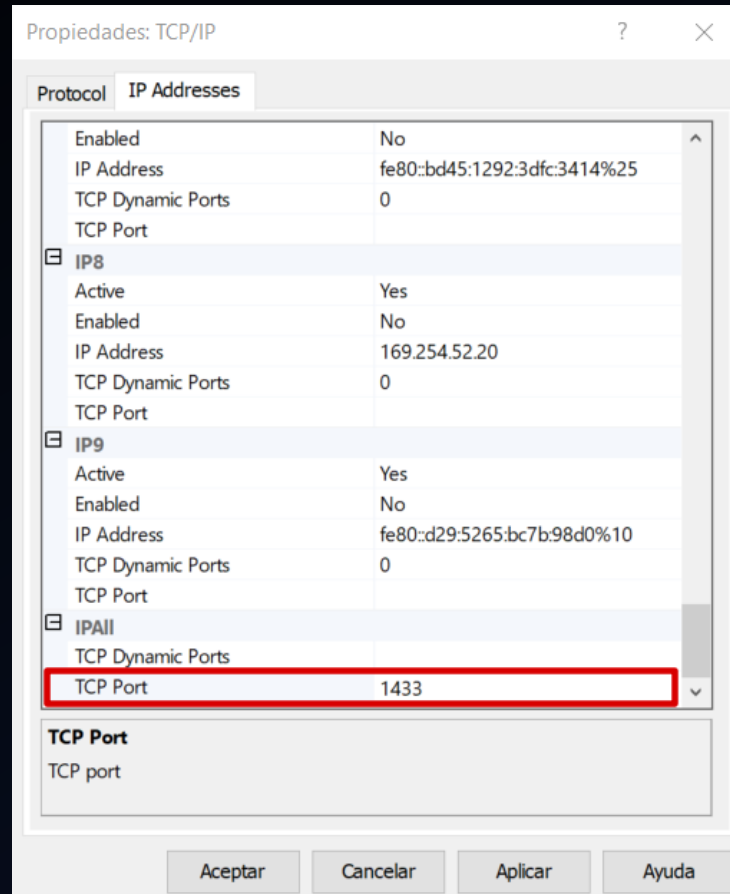
Descripción:

SQL Server admite canalizaciones por protocolos TCP/IP, por nombre o memoria compartida, sin embargo, la buena práctica dicen que debemos usar el mínimo requerido para la continuidad del negocio.

Justificación:

Reduce la superficie de ataques que puede recibir SQL, en algunos casos, puede proteger a la instancia de ataques remotos.

SQL SERVER PORTS



Descripción:

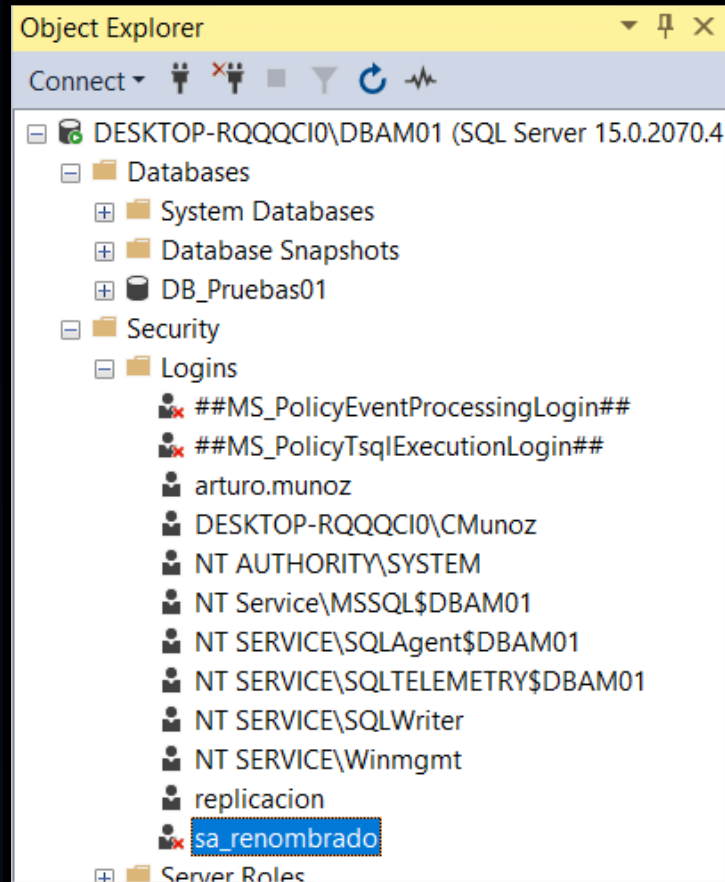
Cuando instalamos una instancia predeterminada de SQL se le asignará el puerto 1433 al servicio Database Engine para la comunicación TCP/IP.

Justificación:

Al tratarse de un puerto por defecto y que a la vez es muy conocido, “no” es recomendable su uso y menos en un entorno productivo.

Lo correcto es que cada instancia de DB maneje su propio puerto TCP/IP dedicado.

USUARIO "SA"



Descripción:

Es un inicio de sesión que debe renombrarse y deshabilitarse ya que es muy conocido y cuenta con privilegios de administrador.

Justificación:

Reduce la probabilidad de recibir ataques de fuerza bruta contra un principio conocido.



CUENTAS DE SERVICIOS SQL

Descripción:

Son aquellos usuarios de AD que se encuentran pegados a los servicios de SQL Server instalados en el servidor.

Justificación:

Siguiendo el privilegio mínimo, la cuenta de los servicios SQL no deben tener más privilegios de los necesarios. Las cuentas NT AUTHORITY\SYSTEM o LocalSystem tienen mayores privilegios a los que son requeridos por SQL.

Name	State	Start Mode	Log On As
 SQL Server (DBAM01)	Running	Automatic	NT Service\MSSQL\$DBAM01
 SQL Full-text Filter Daemon Launcher (DBAM01)	Running	Manual	NT Service\MSSQLFDLauncher\$DBAM01

CCN-STIC-575 – GUÍA DE SEGURIDAD SQL SERVER

Servicio	Recurso	Acceso
MSSQLServer	Instid\MSSQL\backup	Control total
	Instid\MSSQL\binn	Lectura, Ejecución
	Instid\MSSQL\data	Control total
	Instid\MSSQL\FTData	Control total
	Instid\MSSQL\Install	Lectura, Ejecución
	Instid\MSSQL\Log	Control total
	Instid\MSSQL\Repldata	Control total
	130\shared	Lectura, Ejecución
	Instid\MSSQL\Template Data (solo SQL Server Express)	Lectura

Servicio	Grupo de usuarios	Permisos concedidos
Motor de base de datos de SQL Server	Todos los derechos se conceden al SID por servicio. Instancia predeterminada: NT SERVICE\MSSQLSERVER . Instancia con nombre: NT SERVICE\MSSQL\$Nombre_de_instancia .	Iniciar sesión como servicio (SeServiceLogonRight) Reemplazar un token de nivel de proceso (SeAssignPrimaryTokenPrivilege) Omitir comprobación de recorrido (SeChangeNotifyPrivilege) Ajustar las cuotas de la memoria para un proceso (SeIncreaseQuotaPrivilege) Permiso para iniciar el objeto de escritura de SQL Permiso para leer el servicio Registro de eventos Permiso para leer el servicio Llamada a procedimiento remoto

AUDITORÍA DE LOGINS

Este tipo de auditoría nos proporciona trazabilidad en las conexiones a la instancia de base de datos, como también nos permite detectar información crucial en un momento justo (antes que se comprometa la información).

Plus:

Existen formas de monitorear la conexión a la instancia de DB antes que se autorice el acceso, en base a algunos criterios yo puedo autorizar o rechazar la conexión “aumentando la seguridad”.

AUDITORÍA DE EVENTOS

- Nos permite registrar la actividad que tiene diariamente las bases de datos y en base a eso alertar cuando se produce un evento sospechoso.
- SQL Audit permite escribir eventos en algunos niveles:
 - ✓ File
 - ✓ Security log
 - ✓ Application log

CIFRADO TDE

- Es un cifrado transparente de datos que protege los archivos MDF, LDF y BAK.
- Sin el certificado de encriptación y la llave privada del mismo no se puede realizar las siguientes tareas:
 - ✓ Attach database
 - ✓ Restore database

Vuelve inútil el robo de los backup o de la DB en general.

RESUMEN DE VULNERABILIDADES IMPORTANTES

1. Abuso de privilegios.
2. Autenticación débil.
3. Accesos no autorizados.
4. Auditoría débil.
5. Protocolos de base de datos innecesarios.
6. Exposición de datos (backup).
7. Vulnerabilidades de plataforma (SO)
8. SQL Inyección.
9. Denegación de servicio.



TALLER PRÁCTICO



¿PREGUNTAS?