Universidad de Buenos Aires

Facultades de Ciencias Económicas, Ciencias Exactas y

Naturales e Ingeniería

Carrera de Especialización en Seguridad Informática

Trabajo Final

Tema

Seguridad en entornos de bases de datos

Título

“Desarrollo de un esquema de seguridad en SQL Server”

Autor: Ing. Javier Tiebas

Tutor: Ing. Hugo Pagola

Año de presentación 2017
Cohorte 2014-2015

1 Resumen

En este trabajo se presentará el desarrollo de un esquema de configuraciones de seguridad para ser utilizado durante la implementación de un sistema gestor de bases de datos y del sistema operativo sobre el cual va a estar instalado.
Se pretende brindar una forma clara y sencilla para configurar de manera segura un ambiente en el cual residirán de manera conjunta el motor de base de datos y el sistema operativo que lo contiene, teniendo en cuenta las cuestiones más críticas en lo referente a seguridad en la instalación del motor de base de datos y su posterior mantenimiento y administración.
Palabras clave: SQL, SQL Server, DBA, Base de datos, RDMS, backups, 2012 R2, Windows Server, Roles, Schemas, Seguridad, SQL Server 2014.

2 Introducción

La protección de la información es un proceso crítico en las organizaciones desde el momento en que la misma es creada hasta que es destruida, dentro de este proceso la tecnología tiene un papel de suma importancia, especialmente las tecnologías referidas a bases de datos.

Las organizaciones deben proteger sus activos de información, y particularmente las bases de datos, de cualquier tipo de amenaza a la cual se encuentren expuestas. Por lo cual en este trabajo se pretende plantear la generación de un ambiente seguro en cual residan las bases de datos de una organización cuya información debe ser resguardada cumpliendo con los servicios fundamentales que brinda la seguridad de la información, siendo los mismos: confidencialidad, integridad y disponibilidad.

El hecho de implementar mecanismos de seguridad sobre el motor de base de datos es una tarea ardua, si bien existen buenas prácticas de cómo realizar configuraciones en cada plataforma, las mismas no son tenidas en cuenta y muchas veces se cometen omisiones y/o errores de configuración al integrar las diferentes características provistas por cada motor, estos errores u omisiones van desde el uso de contraseñas débiles hasta cuentas de servicios con privilegios demasiado elevados. La mayoría de estos errores u omisiones son introducidos en una etapa temprana, generalmente cuando se instala el motor de base de datos, y muchas veces estas fallas cometidas durante la instalación se mantienen a lo largo del tiempo hasta que finalmente son detectadas a través un una auditoría de seguridad técnica o cuando la información llega a estar comprometida por causa de algún tipo de ataque, en casos como estos se puede decir que la organización toma una postura reactiva en lugar de proactiva en lo referente a la protección de la información.

También es muy común que cada nueva versión de un motor de base de datos que sale al mercado traiga consigo nuevas características relacionadas con la seguridad. Muchas de estas nuevas características no son utilizadas y en caso

de serlo podrían ser implementadas de manera incorrecta ya sea por desconocimiento de su existencia o simplemente por no saber cómo utilizarlas.

Otro punto a tener en cuenta es la especificación de los niveles de seguridad aplicados sobre el sistema operativo en el cual se encontrará alojado el motor de base de datos ya que si los mismos son muy elevados se podría llegar a restringir en forma significativa la funcionalidad de dicho sistema, es por ello que encontrar un balance adecuado entre el nivel de seguridad y la usabilidad del sistema va a ser un punto importante dentro de este trabajo.

La idea de realizar esta guía es para tener un claro panorama sobre cómo implementar la seguridad sobre un motor de base de datos y además como implementar las opciones de seguridad necesarias sobre el sistema operativo donde se encuentre instalado dicho motor, ya que de no considerar esto último estaríamos dejando de lado un factor importante dentro del esquema de seguridad, ya que sólo asegurar el motor de base de datos sin tener en cuenta el sistema operativo, sería algo equivalente a cerrar todas las puertas de una casa pero dejando algunas ventanas abiertas.

3 Recomendaciones generales

Las configuraciones de servidores, base de datos, dispositivos de comunicación y diversos sistemas son las que definen como este conjunto de elementos se deben comportar y actuar. Pero todo este universo fue diseñado para cumplir tareas específicas y no para ser seguros y es por esto que se debe realizar una gestión de la seguridad y decidir que "puertas" y que "ventanas" permanecerán abiertas en nuestros sistemas.

3.1 Nociones de diseño

Existen principios que pueden ser aplicados en el diseño de una estrategia de seguridad para ser aplicada en los servidores y algunos de estos son los sugeridos por Jerome H. Saltzer y Michael Schroeder en su publicación "*The Protection of Information in Computer Systems*" [1]:

1) *Economy of mechanism*: Mantener un diseño tan simple como sea posible.
2) *Fail-safe defaults*: Fallar en forma segura, si se produce una falla, los mecanismos de seguridad deben mantenerse, es preferible perder algo de funcionalidad a perder seguridad.
3) *Complete mediation*: Implementar "intermediarios" en las políticas de acceso.
4) *Open design*: El diseño de los controles de seguridad no debe basarse en el secreto. Los mecanismos de seguridad no deben depender de la ignorancia de los potenciales atacantes.
5) *Separation of privilege*: Implementar la segregación de roles en los usuarios del servidor, para evitar la incompatibilidad de funciones.
6) *Least privilege*: Cada programa y cada usuario debe operar utilizando el mínimo conjunto de privilegios necesario para completar su tarea. Este principio tiene como objetivo limitar el daño que se puede causar por accidente o por error.

7) *Psychological acceptability*: Los usuarios deben entender la necesidad de implementar controles de seguridad. Esto puede implementarse a través de programas de concientización.
8) *Work factor*: Los mecanismos de seguridad deben ser implementados de tal manera que el costo para un atacante exceda el valor de la ganancia que podría obtener en caso de tener éxito.
9) *Compromise recording*: Se deben mantener registros, de manera que si se llegara a concretar una intrusión se pueda obtener evidencia del ataque, esta información podría ser utilizada para identificar el tipo de ataque que fue realizado, métodos y *exploits* utilizados por el atacante.

3.2 Categorización de la información

Determinar que tanto hay que proteger el servidor es algo que se debe definir desde un primer momento, ya que el nivel de seguridad seleccionado será el que condicione hasta donde se va a llegar en cuanto a las medidas de seguridad a implementar. El tipo de información que será protegida será, por lo general, el factor determinante en la selección del nivel de seguridad, así como también lo serán los servidores que se encuentren dentro de la misma red.
La *Federal Information Processing Standards* (FIPS) en su publicación "Standards for sercurity categorization of federal information and information systems" [2], establece una serie de criterios para determinar una categoría de seguridad y asignársela a un servidor.
La categorización que realiza FIPS se basa en el impacto a cada uno de los tres pilares de seguridad de la información: Confidencialidad, Integridad y Disponibilidad.

Los niveles que se definen en [2] son los siguientes:

"El impacto potencial es BAJO si la pérdida de confidencialidad, integridad o disponibilidad de la información tiene un efecto adverso limitado en las operaciones de la organización, sus activos, o los individuos. Un efecto adverso

que podría generar la pérdida de confidencialidad, integridad o disponibilidad podría ser:

- Provocar una degradación de la capacidad de operación de la organización siendo capaz de realizar sus funciones primarias, pero la efectividad de sus funciones se ve reducida notablemente.
- Daños menores a los activos de la organización, pérdidas financieras menores, o daños menores a los individuos.

El impacto potencial es MODERADO si la pérdida de confidencialidad, integridad o disponibilidad de la información tiene un efecto adverso sobre las operaciones de la organización, sus activos, o individuos. Un efecto adverso grave que podría generar la pérdida de confidencialidad, integridad o disponibilidad podría ser:

- Provocar una degradación importante de la capacidad de operación de la organización, siendo capaz de realizar sus funciones primarias, pero la efectividad de sus funciones se ve reducida notablemente.
- Daño significativo a los activos de la organización, pérdidas financieras significativas o daño significativo a las personas, pero no la pérdida de vidas o lesiones graves que amenazan la vida.

El impacto potencial es ALTO si la pérdida de confidencialidad, integridad o disponibilidad de la información tiene un efecto adverso grave o catastrófico en las operaciones de la organización, sus activos, o individuos. Un efecto adverso grave o catastrófico que podría generar la pérdida de confidencialidad, integridad o disponibilidad podría ser:

- Provocar una degradación severa o pérdida de la capacidad de operación de la organización con lo cual ésta no es capaz de realizar una o más de sus funciones primarias.
- Daño mayor a los activos de la organización, mayor pérdida financiera o daño grave o catastrófico para personas con pérdida de vidas o lesiones graves que amenazan la vida.”

3.3 Niveles de seguridad

El propósito de esta sección es asistir a las organizaciones en la comprensión de la importancia de la realización de actividades para asegurar y mantener seguros a lo largo del tiempo los servidores que proveen algún tipo de prestación a la organización, también se tiene como objetivo brindar recomendaciones para seleccionar, implementar y mantener controles de seguridad adecuados.

Las organizaciones deben asegurarse de que los sistemas operativos de sus servidores sean correctamente instalados, configurados y logren cumplir con los requisitos de seguridad establecidos en “las buenas prácticas” así como también en las políticas de seguridad de la organización.

Es recomendable tener siempre presente la premisa que dice que “el primer paso para obtener un servidor seguro es asegurar el sistema operativo subyacente”. Partiendo de esta premisa entendemos que muchos problemas de seguridad podrían evitarse si el sistema operativo fuera configurado correctamente desde el momento en que este es instalado en el servidor. Es un hecho común que una instalación se realice con las configuraciones que vienen por defecto, tanto para el hardware como para el software. Estás configuraciones por defecto son provistas por los fabricantes del hardware y del software con el objetivo de facilitar la instalación y el uso de sus productos, pero esto muchas veces se hace a expensas de la seguridad. Cada administrador de sistemas debe configurar cada uno de sus servidores para reflejar los requisitos de seguridad de la organización y reconfigurarlos a medida que cambien los mismos.

Una buena práctica es la definición de estándares de configuración los cuales pueden incluir un *checklist* para validar la configuración de seguridad en cada instalación de un nuevo servidor y también para cuando se realiza algún cambio, o simplemente para realizar revisiones rutinarias con el objetivo de verificar que las configuraciones no fueron alteradas.

3.4 Planificación de la instalación

Un aspecto clave a la hora de comenzar la instalación de un servidor, es tener un plan detallado del proceso que se llevará adelante, tener un plan brindará una guía que permitirá tener un servidor tan seguro como se haya planificado.

Algunos puntos a ser considerados en la planeación de la seguridad en el servidor son los mencionados por Julia Allen en “Securing Network Servers” [3]:

- Identificar el propósito del servidor
    - Qué tipo de información va a almacenar.
    - Qué tipo de información va a procesar o transmitir.
    - Cuáles son los requerimientos de seguridad para los hosts relacionados.
    - Que otros servicios va a proveer el servidor (se recomienda que sea de propósito específico)
    - En qué lugar de la red va a estar ubicado (red interna, DMZ, etc.).
- Identificar los servicios de red que proveerá.
- Identificar usuarios y categorías de usuarios que tendrá el servidor y los hosts relacionados.
- Determinar como el servidor será administrado (localmente, remotamente).
- Decidir cómo serán autenticados los usuarios.
- Determinar cómo se realizarán los accesos a los recursos del servidor.

Para la administración de servidores y ambientes es recomendable desarrollar configuraciones estándar para cada tipo de servidor dentro de la organización, ya que un solo servidor incorrectamente configurado puede llegar a comprometer otros servidores dentro de la red. Esta recomendación tiende a ser de carácter obligatorio cuando la organización posee un gran número de servidores.

3.5 Aseguramiento del sistema operativo

Implementar la seguridad por capas es una estrategia ampliamente utilizada, por lo cual una buena configuración del sistema operativo ayudará a fortalecer las capas exteriores en nuestro esquema de seguridad.

Los pasos básicos necesarios para una correcta configuración de seguridad del sistema operativo según el NIST en "Guide to General Server Security" [4] son listados a continuación:

Parchear y actualizar el Sistema Operativo:

- Crear, documentar e implementar un proceso de parcheo;
- identificar vulnerabilidades y los parches correspondientes;
- mitigar vulnerabilidades temporalmente si es necesario y si es posible, hasta que los parches estén disponibles, testeados e instalados;
- instalar correcciones permanentes (parches, actualizaciones, etc.);
- mantener los servidores desconectados de las redes o solamente conectarlos a una red que se encuentre aislada hasta que todos los parches hayan sido instalados, y todos los pasos de configuración se hayan realizado;
- colocar los servidores en una VLAN u otro segmento que restrinja las acciones que pueden realizar los hosts conectados y que comunicaciones pueden alcanzar a dichos hosts.

Los administradores no deberían aplicar los parches a servidores de producción sin antes haberlos probado en servidores de prueba idénticamente configurados, ya que los parches podrían causar problemas inesperados al servidor.

Reforzar la configuración del Sistema Operativo

- **Desinstalar o deshabilitar los servicios innecesarios**

  Se deberán desinstalar o deshabilitar los servicios, aplicaciones o protocolos de red innecesarios, aunque es preferible desinstalar los servicios

innecesarios en lugar de deshabilitarlos, ya que un ataque podría reactivarlos o también podrían activarse de forma involuntaria por un error.

Idealmente un servidor debería ser dedicado a un único propósito. Al configurar el sistema operativo hay que desinstalar todos los servicios, aplicaciones y protocolos de red que no son requeridos y deshabilitar componentes innecesarios que no se hayan podido desinstalar. De ser posible, instalar la configuración mínima del Sistema Operativo, e instalar, desinstalar y deshabilitar servicios, aplicaciones y protocolos de red a medida que sea necesario.

Los siguientes son ejemplos de servicios y aplicaciones que comúnmente deben ser desinstaladas (o deshabilitadas) si no son requeridas: servicios para compartir archivos e impresoras como Windows Network Basic Input/Output System (NetBIOS), Network File System (NFS), FTP; servicios para redes inalámbricas; programas de control y acceso remoto, particularmente aquellos que no cifran sus comunicaciones como Telnet, si el uso de un programa de acceso remoto es absolutamente necesario y este no cifra las comunicaciones, las mismas podrían ser realizadas a través de un túnel sobre un protocolo que provea de un cifrado de los datos como SSH, IPsec; servicios de directorio como LDAP, Network Information System (NIS); servidores y servicios web; servicios de email como SMTP; compiladores; herramientas de desarrollo; herramientas de administración de red como Simple Network Management Protocol (SNMP).

- **Configurar la autenticación de usuarios en el Sistema Operativo**

Para obtener una autenticación apropiada se recomienda seguir los siguientes pasos mencionados por Julia Allen en [3].

1. Eliminar o desactivar cuentas por defecto que no son necesarias. La configuración por defecto del sistema operativo incluye a menudo las cuentas de invitados (con y sin contraseñas), cuentas de administrador y cuentas asociadas servicios locales y de red. Quitar o desactivar cuentas innecesarias para eliminar su uso por parte de intrusos,

incluyendo cuentas de invitados en los servidores que contienen información sensible. Si se tiene la obligación de mantener una cuenta de invitado o grupo, restringir su acceso y cambiar la contraseña de acuerdo a la política de contraseñas. Para las cuentas por defecto que se necesiten mantener, se deberán renombrar (cuando sea posible y particularmente para las cuentas de administrador) y cambiar contraseñas para que sean coherentes con la política de contraseñas.

2. Desactivar cuentas no interactivas. Desactivar cuentas (y las contraseñas asociadas) que deben existir, pero no requieren un login interactivo.
3. Crear grupos de usuarios en el servidor. Asignar usuarios a los grupos adecuados. A continuación, asignar los derechos de los grupos, como se documenta en el plan de implementación. Este enfoque es preferible a la asignación de derechos a los usuarios de manera individual.
4. Crear sólo las cuentas de usuario necesarias. Permitir el uso de cuentas compartidas sólo cuando no existan alternativas viables. Tener cuentas de usuario comunes para los administradores quienes son también usuarios del servidor.
5. Configurar la sincronización automática. Algunos protocolos de autenticación, como Kerberos, no funcionan si la diferencia de tiempo entre el equipo cliente y el servidor de autenticación es significativa, por lo que los servidores que utilizan estos protocolos deben ser configurados para sincronizar automáticamente la hora del sistema con un servidor de tiempo confiable. Normalmente, el servidor de tiempo es interno a la organización y utiliza el Protocolo de Tiempo de Red (*NTP-Network Time Protocol*) para la sincronización; existen servidores NTP disponibles en Internet.
6. Comprobar la política de contraseñas de la organización. Los siguientes son algunos elementos que podrían incluirse en una política de contraseñas:
    - Longitud: Establecer una longitud mínima para las contraseñas.

- Complejidad: Mezclar caracteres, por ejemplo, forzar el uso de contraseñas que contengan letras mayúsculas, minúsculas y caracteres no alfabéticos, y que no contengan palabras "de diccionario".
- Envejecimiento: cuánto tiempo una contraseña puede permanecer sin ser cambiada. Muchas políticas requieren que los usuarios y administradores cambien periódicamente sus contraseñas. En tales casos, la frecuencia debe ser determinada por la longitud forzada y la complejidad de la contraseña, la sensibilidad de la información protegida, y el nivel de exposición de contraseñas. Debe considerarse la posibilidad de imponer una duración mínima de envejecimiento para evitar que los usuarios cambien repetidas veces la contraseña para limpiar el historial y evitar la restricción de reutilización de contraseñas.
- Reutilización: Refiere a si una contraseña puede ser reutilizada. Algunos usuarios tratan de evitar el requisito de envejecimiento de contraseña cambiando la contraseña por una que han utilizado anteriormente. Si la reutilización está prohibida por política de la organización, es beneficioso, de ser posible, asegurarse de que los usuarios no puedan cambiar sus contraseñas simplemente añadiendo caracteres al principio o al final de sus contraseñas originales (por ejemplo, si la contraseña original era "password" y se cambia a "1password " o " password1").
- Autoridad responsable, esto refiere a quién está autorizado a cambiar o restablecer las contraseñas y qué tipo de prueba se requiere antes iniciar cualquier cambio.
- Seguridad de contraseña. Se refiere a cómo se deben asegurar, evitar almacenar contraseñas sin cifrar en el servidor, y requerir a los administradores utilizar diferentes contraseñas para sus diferentes cuentas en el servidor.

- Configurar los equipos para prevenir la posibilidad de adivinar la contraseña. Es relativamente fácil para un usuario no autorizado tratar de obtener acceso a un servidor mediante el uso de herramientas de software automatizadas para intentar adivinar las contraseñas.
  El Sistema Operativo se deberá configurar para aumentar el período entre intentos de conexión con cada intento fallido. Si eso no es posible, la alternativa es negar el acceso después de un número limitado de intentos fallidos (por ejemplo, tres). Normalmente, la cuenta se bloquea por un período de tiempo (por ejemplo, 30 minutos) o hasta que un usuario con la suficiente autoridad la reactiva.
  La elección de negar el acceso es otra situación que requiere que el administrador del servidor tome una decisión que equilibre la seguridad y la comodidad. La implementación de esta recomendación puede ayudar a prevenir algunos tipos de ataques, pero también puede permitir a un atacante utilizar los intentos fallidos de conexión para denegar el acceso de usuarios, lo que resulta en una condición de denegación de servicio conocida como DoS por sus siglas en inglés. El riesgo de DoS de bloqueo de cuentas es mucho mayor si el servidor es accesible desde el exterior y un atacante sabe o puede suponer la convención de nombres utilizada por la organización, lo cual le permitiría adivinar los nombres de cuenta.
  Los intentos de conexión fallidos de red no deben impedir que un usuario administrador no pueda realizar un inicio de sesión en la consola. Nótese que todos intentos de conexión fallidos, ya sea a través de la red o de la consola, deben ser registrados.
  Si el servidor no será administrado de forma remota, se deberá deshabilitar este permiso de todas las cuentas de administración.

- **Establecimiento de controles sobre los recursos del servidor.**

Al establecer cuidadosamente los controles de acceso y negando a los usuarios el acceso no autorizado, es posible reducir las brechas de seguridad intencionales y no intencionales que puedan existir. El servidor provee la capacidad para especificar privilegios de acceso de forma individual para archivos, directorios, dispositivos y otros recursos computacionales.

- **Instalar y configurar controles de seguridad adicionales**

Muchos sistemas operativos no incluyen los controles de seguridad necesarios para garantizar la seguridad de los servicios y aplicaciones adecuadamente. Por lo tanto, se deberá instalar, configurar y mantener software adicional para proporcionar los controles que faltan. Se recomiendan implementar los siguientes controles a través de la instalación de diferentes tipos de software como:

- software anti-malware: antivirus, anti-spyware, y detectores de rootkits;
- HIDS- Sistemas de detección de intrusos para hosts;
- software para gestión de vulnerabilidades y administración de parches.

3.6 Testeos regulares de seguridad del servidor

Para mantener a lo largo del tiempo una configuración de seguridad tangible es recomendable testear de forma regular los controles de seguridad implementados en el servidor, los métodos más comunes utilizados incluyen el escaneo de vulnerabilidades y pruebas de intrusión.

3.6.1 Escaneos de vulnerabilidades

Los escaneos de vulnerabilidades se realizan de forma automática con herramientas que identifican vulnerabilidades y configuraciones incorrectas del servidor. Estos escaneos ayudan a identificar versiones de software desactualizadas, parches que faltan aplicar o actualizaciones del sistema operativo faltantes.

3.6.2 Pruebas de intrusión

El *Committee on National Security Systems* en el *National Information Assurance Glossary* (*CNSS Instruction N° 4009*) define prueba de intrusión como: "*A test methodology in which assessors, typically working under specific constraints, attempt to circumvent or defeat the security features of an information system*"

En el caso de realizar pruebas de intrusión sobre los servidores de producción se deberá analizar el impacto que tendrán los mismos sobre cada servidor, ya que, por ejemplo, la prueba podría causar una denegación de servicio al intentar iniciar sesión con usuarios y contraseñas que comúnmente vienen por defecto en diferentes servicios. También hay que considerar el tipo de información que contienen los servidores ya que ciertas pruebas podrían llegar a exponer información sensible a personas no autorizadas, en un caso así deberían realizarse las pruebas sobre servidores de testing habiendo realizado una previa sanitización de los datos siguiendo los pasos recomendados por NIST en "Guidelines for Media Sanitization" [5], tema que se tratará más adelante en este trabajo.

Si se implementa un servidor de testeo, el mismo deberá tener hardware y software idéntico al del servidor de producción, al igual que sus distintas configuraciones, y deberá estar dentro de una red interna a la organización. Este tipo de pruebas tiene varias ventajas, entre las cuales podemos destacar las pruebas de parches y de service pack antes de aplicarlos en producción, tener un ambiente de prueba para nuevas aplicaciones a instalar en el servidor, testeo de diferentes configuraciones.

3.7 Administración remota

La administración remota de un servidor se debe permitir sólo tras una cuidadosa consideración de los riesgos según [2]. El riesgo que implica este tipo de administración varía considerablemente dependiendo de la ubicación del servidor en la red. Para un servidor que se encuentra detrás de un firewall, la administración se puede implementar relativamente de forma segura desde la red interna. La administración remota generalmente, no se debe permitir desde un

host situado fuera de la red de la organización a menos que se realice desde un equipo controlado por la organización a través de un acceso remoto a la organización, como una VPN.
Si una organización determina que es necesario administrar de forma remota un servidor, es conveniente seguir los siguientes pasos para garantizar que la administración se implementa de manera tan segura como sea posible:

- Utilizar un mecanismo de autenticación fuerte (autenticación de dos factores o más) como recomienda el estándar PCI-DSS [6];
- Restringir que hosts se pueden utilizar para administrar de forma remota el servidor.
    - Restringir los usuarios autorizados,
    - restringir por dirección IP (no el nombre de host),
    - limitar a los hosts de la red interna, o los que utilizan las empresas para acceder de forma remota.
- Utilizar protocolos seguros que pueden proporcionar el cifrado de las contraseñas y los datos que se transmiten. No utilizar protocolos menos seguros (por ejemplo, telnet, FTP, NFS, HTTP) a menos que sea absolutamente necesario, en ese caso utilizar un túnel a través de un protocolo de cifrado, como SSH, SSL o IPsec.
- Hacer cumplir el concepto de “mínimo privilegio” en la administración remota (por ejemplo, tratar de minimizar los derechos de acceso remoto a las cuentas de administración).
- No permitir la administración remota desde Internet a menos que sea a través de mecanismos de seguridad fuertes, tales como VPNs.
- Utilizar protocolos de administración remota que soporten la autenticación de servidor para prevenir los ataques man-in-the-middle.
- Cambiar las cuentas por defecto o contraseñas de la utilidad de administración remota o aplicación.

4 Herramientas para Windows Server 2012 R2

Hay muchas recomendaciones sobre como reforzar la seguridad, como por ejemplo, detener servicios que no son necesarios o desactivar características del sistema operativo que no serán utilizadas, y aunque cada versión de Windows Server ya está preparada de manera predeterminada con configuraciones por defecto más seguras, es recomendable tener total conocimiento sobre qué servicios se encuentran activos en cada rol de servidor que se ha definido, que permisos de acceso tiene un directorio determinado, que usuarios existen en el sistema, etc., y para esto existen diversas herramientas que nos van a dar soporte en esta tarea.

A continuación se detallan algunas de las aplicaciones que pueden utilizarse para reforzar la seguridad de nuestro servidor Windows Server 2012 R2.

4.1 Security Configuration Wizard [7]

“El Asistente para configuración de seguridad (SCW) le guía a través del proceso de creación, edición, aplicación o reversión de una directiva de seguridad. Una directiva de seguridad que se crea con SCW es un archivo .xml que, cuando se aplica, configura servicios, seguridad de red, valores de registro específicos y directiva de auditoría. SCW es una herramienta basada en roles: puede usarla para crear una directiva que habilite servicios, reglas de firewall y la configuración necesaria para que un servidor seleccionado realice roles específicos, como un servidor de archivos, un servidor de impresión o un controlador de dominio.”

El Asistente para Configuración de Seguridad [7] (SCW por sus siglas en inglés) es una herramienta que reduce la superficie de ataque y ayuda a crear políticas de seguridad que se basen en la funcionalidad mínima requerida por las funciones del servidor.

Esta herramienta es utilizada como ayuda en la creación de políticas de seguridad las cuales podrán ser aplicadas en diferentes roles de servidor, estas políticas pueden ser utilizadas para configurar servicios (habilitar o deshabilitar), auditoría y entradas de registro (registry).

Su funcionalidad se basa principalmente en revisar los puertos que se encuentran abiertos en el servidor, por eso para su ejecución es necesario asegurarse que los servicios que requiere el servidor se encuentren en ejecución, ya que de lo contrario los mismos no serían detectados por la aplicación.

Como resultado final se obtiene un archivo xml con todas las configuraciones establecidas, las cuales pueden ser aplicadas al finalizar el asistente de configuración o pueden aplicarse en otro momento volviendo a ejecutar la herramienta.

4.2 Microsoft Baseline Security Analyzer 2.3

De acuerdo con el artículo publicado por Microsoft [8] "Microsoft Baseline Security Analyzer (MBSA) es una herramienta fácil de usar diseñada para los profesionales de TI que ayuda a las pequeñas y medianas empresas a determinar su estado de seguridad según las recomendaciones de seguridad de Microsoft y ofrece orientación de soluciones específicas."

Esta herramienta ayuda a mejorar el proceso de administración de seguridad para detectar los errores más comunes de configuración de seguridad, actualizaciones de seguridad que falten en los sistemas, qué directorios se comparten y con quién, posibles vulnerabilidades en Windows, IIS, SQL Server, contraseñas débiles, etc. MBSA permite realizar escaneos de servidores dentro de la misma red.

4.3 Microsoft Security Compliance Manager [9]

"Security Compliance Manager (SCM) es una herramienta gratuita del equipo de Aceleradores de soluciones de Microsoft que le permite configurar y administrar rápidamente los equipos de su entorno y su nube privada mediante la directiva de grupo y el Administrador de configuración de Microsoft System Center."

Esta versión muestra un listado de líneas base con una estructura de árbol y agrupadas por diferentes productos (Windows, Exchange, etc.). Estas líneas base no pueden modificarse pero si pueden copiarse para generar políticas personalizadas. Las mejoras en esta herramienta se basan en la

retroalimentación de las experiencias de los clientes que la utilizan en todo el mundo.

4.4 Dumpsec

Esta herramienta fue desarrollada por SomarSoft, y en su sitio web es descrita de la siguiente manera [10]: “DumpSec de SomarSoft es un programa de auditoría de seguridad para Microsoft Windows® NT / XP / 200x. Vuelca los permisos (DACL) y la configuración de auditoría (SACL) para el sistema de archivos, el registro, las impresoras y los recursos compartidos en un formato conciso y legible, de modo que los agujeros en la seguridad del sistema sean fácilmente evidentes. DumpSec también descarga la información de usuarios, grupos y replicación.” Esta herramienta facilita llevar un control sobre permisos de usuarios, y los grupos a los cuales pertenecen, también permite descargar las políticas del sistema, los derechos de usuarios y servicios que se están ejecutando. Se puede ejecutar de manera remota en equipos conectados a la misma red.

4.5 AccessEnum

Esta es una herramienta desarrollada por Microsoft, la cual es definida como: “AccessEnum le ofrece una vista completa de su sistema de archivos y la configuración de seguridad del Registro en cuestión de segundos, por lo que es la herramienta ideal para ayudarle en los agujeros de seguridad y bloquear los permisos cuando sea necesario.
AccessEnum utiliza la API de seguridad estándar de Windows para rellenar una vista con información de acceso de lectura, escritura y denegación.”[11]
Esta herramienta nos permite ver los permisos efectivos que de cada carpeta y archivo. Si bien AccessEnum no permite modificar dichos permisos, nos va a permitir identificar quienes tienen acceso a carpetas o archivos críticos, como podrían ser los archivos de log, o los directorios donde se encuentra instalado el SQL Server.

5 SQL Server

5.1 Autenticación y autorización

En esta sección se expondrán recomendaciones relacionadas a los mecanismos de autenticación y autorización de SQL Server y las cuentas recomendadas para ser utilizadas por los servicios.

De acuerdo con el artículo de Microsoft “Autenticación en SQL Server” [12]:“SQL Server admite dos modos de autenticación, el modo de autenticación de Windows y el modo mixto.

- La autenticación de Windows es el modo predeterminado, y a menudo se denomina seguridad integrada debido a que este modelo de seguridad de SQL Server está integrado estrechamente en Windows. Para iniciar sesión en SQL Server, se confía en las cuentas de usuario y grupo específicas de Windows. Los usuarios de Windows que ya hayan sido autenticados no tienen que presentar credenciales adicionales.
- El modo mixto admite la autenticación tanto de Windows como de SQL Server. Los pares de nombre de usuario y contraseña se mantienen en SQL Server.”

De ser posible se recomienda utilizar la autenticación de Windows ya que la misma provee un mecanismo más robusto de autenticación que la autenticación de SQL Server.

El valor por defecto es autenticación de Windows.

Un método para verificar el tipo de autenticación que se ha implementado en el servidor es ejecutar el siguiente comando:

xp_loginconfig 'login mode';

Como resultado se pueden obtener dos valores de config_value:

- “Windows NT Authentication”(fig 1), indica autenticación de Windows
- “Mixed” (fig 2), indica autenticación mixta.

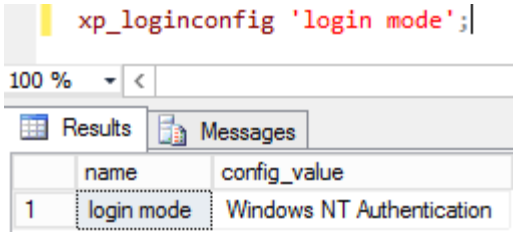

Fig1

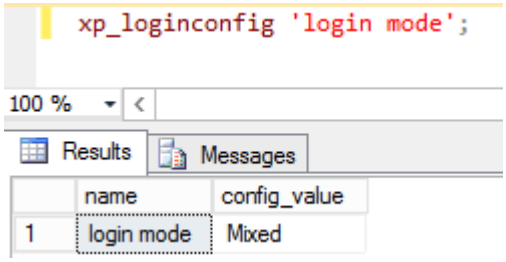

Fig2

También puede utilizarse la siguiente instrucción:

SELECT SERVERPROPERTY('IsIntegratedSecurityOnly')

La cual devolverá un valor de 0 para el caso de autenticación de Windows y un valor de 1 para el caso de autenticación mixta.

5.2 Tipos de cuentas para servicios SQL Server

Según las buenas prácticas es conveniente definir las cuentas que serán utilizadas por los servicios de SQL Server con los menores niveles de derechos posibles.

Como recomienda Microsoft en su artículo “Configurar los permisos y las cuentas de servicio de Windows“[13] “Utilice cuentas de tipo MSA o cuentas virtuales cuando sea posible. Cuando MSA y cuentas virtuales no son posibles, utilice una cuenta de usuario o de dominio con privilegios bajos en lugar de una cuenta compartida de servicios de SQL Server. Utilice cuentas separadas para los diferentes servicios de SQL Server. No conceder permisos adicionales a la cuenta de servicio de SQL Server o los grupos de servicio. Los permisos se otorgan a través de la pertenencia a grupos u otorgados directamente a un servicio SID, donde un servicio SID se encuentre soportado.”

A continuación se detallan los distintos tipos de cuentas recomendados para los servicios de SQL Server 2014 de acuerdo con [13]:

5.2.1 Las cuentas MSA y cuentas virtuales

Cuando se utilizan cuentas de usuario o de dominio para ejecutar servicios nos encontramos con el problema de la gestión de las contraseñas, dado que hay que recordar cuando se vence la contraseña o en su defecto configurar la cuenta para que la misma no se caduque, está última es una práctica bastante común pero no recomendable, dado que si no se toman precauciones la cuenta podría sufrir un ataque de adivinación por fuerza bruta para obtener la contraseña.

Para superar este problema podemos utilizar las cuentas MSA (Managed Service Accounts) y las cuentas virtuales. Este tipo de cuentas introducidas por Microsoft en Windows 7 y Windows Server 2008R2 permiten una gestión automática de las contraseñas. Las cuentas MSA son cuentas de dominio mientras que las cuentas virtuales son cuentas locales. La instalación por defecto del SQL Server 2014 genera cuentas de tipo virtuales con la siguiente nomenclatura: NT SERVICE\<servicename>.

Otra alternativa para las cuentas de servicio podría ser utilizar cuentas de usuario locales, cuentas de servicio locales, cuentas de servicio de red o la cuenta System local.

“La cuenta de servicio administrado (MSA) está diseñada para proporcionar aplicaciones como SQL Server o Exchange con:

- Gestión automática de contraseñas, que puede aislar mejor estos servicios de otros servicios en el equipo.
- Gestión simplificada de nombre principal de servicio (SPN), que permite a los administradores de servicios establecer SPN en estas cuentas. Además, la administración de SPN puede delegarse a otros administradores.” [14]

“Las cuentas virtuales emulan la creación de muchas instancias únicas de la cuenta de servicio de red, por lo que cada servicio se ejecuta con su propia instancia de servicio de red que tiene el mismo nombre que el servicio. Estas

instancias únicas de servicio de red facilitan mucho la auditoría y el seguimiento."[15]

5.2.2 Cuentas de usuario locales

Cuando el servidor donde se encuentra instalado el SQL Server no es parte de un dominio se puede utilizar una cuenta de usuario local de Windows, dicha cuenta no debe poseer permisos de administrador.

5.2.3 Cuenta de servicio local

Este tipo de cuenta viene integrada con el sistema operativo Windows y tiene el mismo nivel de acceso a recursos y objetos que las cuentas pertenecientes al grupo "Usuarios".

Los servicios que se ejecutan como la cuenta de servicio local tienen acceso a recursos de la red local con una sesión nula sin credenciales. Hay que tener en cuenta que la cuenta de servicio local no es compatible con los servicios del Agente SQL Server o el servicio de SQL Server. El nombre real de la cuenta es NT AUTHORITY \ SERVICIO LOCAL.

5.2.4 Cuenta de servicio de red

La cuenta de servicio de red es una cuenta integrada de Windows que tiene acceso a los recursos y objetos que no sean miembros del grupo "Usuarios". El formato de la cuenta es <nombreDominio> \ <equipo> $. El nombre real de la cuenta es NT AUTHORITY \ NETWORK.

5.2.5 Cuenta System local

Esta cuenta viene integrada en Windows y tiene los privilegios más elevados dentro del servidor.

El nombre real de la cuenta es NT AUTHORITY \ SYSTEM.

5.2.6 Cuenta Guest

En SQL Server existe la cuenta de usuario Guest, la cual se utiliza para permitir el acceso a cualquier login del servidor que no esté mapeado a una de base de datos específica. Y es por esto que al revocar el permiso de conexión para el

usuario Guest se asegurará que dicho inicio de sesión no sea capaz de acceder a la información de la base de datos sin darle un acceso explícito.

Ejecutando el siguiente código se podrá verificar si la cuenta Guest tiene permiso de connect sobre la base de datos:

```
USE [database_name];
GO
SELECT DB_NAME() AS DBName, dpr.name, dpe.permission_name FROM sys.database_permissions dpe JOIN sys.database_principals dpr ON dpe.grantee_principal_id=dpr.principal_id WHERE dpr.name='guest' AND dpe.permission_name='CONNECT';
```

El permiso puede ser revocado ejecuntado la siguiente sentencia:

```
USE database_name
REVOKE CONNECT FROM GUEST
```

5.3 Políticas de contraseñas

A continuación se darán recomendaciones sobre políticas de contraseñas que podrán ser aplicadas al SQL Server para reforzar la seguridad.

5.3.1 Establecer la opción de cambio obligatorio de contraseña

En el caso de utilizar la autenticación de SQL Server, la opción de cambio obligatorio de contraseña es utilizada para forzar al usuario propietario del nuevo login o uno ya existente a crear una nueva contraseña que sólo él va a conocer, esto se debe a que se trata de evitar que dicha cuenta sea utilizada por otras personas para hacer un mal uso de la misma dado que la contraseña para el primer inicio es establecida por el usuario administrador de cuentas que creó el login y por lo cual tiene conocimiento de dicha contraseña.

Para verificar si un usuario tiene activo el cambio de contraseña obligatorio basta con acceder a las propiedades de login y ver si tiene la opción de “cambio de contraseña en el próximo login” activada.

Si se quiere forzar el cambio de contraseña de un login se puede utilizar el siguiente código como se especifica en el artículo de Microsoft referido a la modificación de los logins en SQL Server “ALTER LOGIN (Transact-SQL)“ [16]

```
ALTER LOGIN login_name WITH PASSWORD = password_value MUST_CHANGE;
```

Al generar un nuevo login o al realizar un blanqueo de contraseña es recomendable utilizar siempre una primera contraseña diferente, ya que de utilizar siempre la misma primera contraseña un usuario con conocimiento de la existencia de nuevos logins podría intentar acceder con la contraseña conocida.

5.3.2 Establecer la opción de expiración de contraseña

Si se utiliza la autenticación de SQL Server se recomienda establecer la opción de expiración de contraseña al crear el login o modificarlo, está opción tomará los mismos valores establecidos en la política de seguridad de Windows. De acuerdo con Microsoft en la nota de políticas de seguridad [17] “Es una práctica de seguridad tener contraseñas que expiran cada 30 a 90 días, dependiendo del entorno. De esta manera un atacante tiene una cantidad limitada de tiempo para descifrar la contraseña de un usuario y tener acceso a los recursos de red.”

Para verificar si un usuario tiene activa la opción de expiración de contraseña basta con acceder a las propiedades de login y ver si tiene la opción de “Forzar expiración de contraseña” activada o también realizando la siguiente consulta:

select name, is_expiration_checked from sys.sql_logins, el valor “0” en el login indica que no está activada.

Microsoft en su artículo “Directivas de contraseñas” [18] explica que si se quiere forzar expiración de contraseña de un login se puede utilizar el siguiente código:

```
ALTER LOGIN [login_name] WITH CHECK_EXPIRATION = ON;
```

5.3.3 Establecer la opción de aplicar la política de contraseña

Si se seleccionó la autenticación por SQL Server, al seleccionar la opción “Aplicar política de contraseña” se aplica a SQL Server la misma política de contraseñas utilizada en Windows.

El parámetro especifica que el login de SQL Server debe cumplir con la política de contraseñas de Windows, la cual a su vez incluye las políticas de bloqueo de cuenta.
Se puede verificar a través de la siguiente consulta:
select name, is_policy_checked from sys.sql_logins, el valor “0” en el login indica que no está activada.
Para activarla se debe ejecutar la siguiente sentencia de acuerdo con [18]:

```
ALTER LOGIN [login_name] WITH CHECK_POLICY = ON;
```

Es importante tener en cuenta que esta política no obliga a cambiar una contraseña existente que no cumple con los requisitos de la política. Por lo tanto también se recomienda forzar el cambio de contraseña del login en el siguiente inicio.

5.4 Protección de datos

Para proteger una base de datos se pueden tomar varias precauciones, como asegurar el sistema operativo del servidor, la red donde se encuentra alojado y el perímetro de esa red a través de un firewall, todas estas opciones son algo que debemos llevar a cabo, pero además no hay que olvidar que una tarea común y obligatoria sobre las bases de datos es la de realizar bakcups, dichos backups pueden ser alojados en cintas o discos dentro o fuera del mismo servidor. Pero ¿Qué pasaría si las cintas donde se alojan esos backups son robadas?, o si son depositadas en repositorios sobre los cuales tienen acceso diferentes usuarios. Restaurar una base de datos es una tarea simple, y una vez completada nos da acceso a toda la información que dicha base contiene por lo cual la protección de todos los backups que se realizan tiene que estar contemplada en el plan de seguridad. También es muy común que los desarrolladores y testers requieran de datos lo más semejantes a lo que se utilizará en un ambiente productivo, por lo que muchas veces la información de las bases de datos productivas termina en ambientes menos seguros como son los de testing y desarrollo sin que previamente se le realice una sanitización a dichos datos.

Según una clasificación hecha por John Magnabosco en “Protecting SQL Server Data” [19] los datos dentro de nuestra base de datos pueden clasificarse dentro de dos grupos: datos en tránsito y datos almacenados. “Los datos en reposo se refieren a datos que se almacenan, archivan o residen en medios de copia de seguridad. Datos en tránsito se refiere a datos que están atravesando una red, o que residen en la memoria. Ambos estados de datos tienen sus preocupaciones de seguridad y métodos de mitigación de amenazas.”

El estándar PCI-DSS [6] hace una clara diferencia en esta clasificación en lo que refiere a datos en reposo y a datos en tránsito en su requisito 3: “Proteja los datos del titular de la tarjeta que fueron almacenados. Los métodos de protección como el cifrado, el truncamiento, el ocultamiento y la refundición son importantes componentes de la protección de datos del titular de la tarjeta. Si un intruso viola otros controles de seguridad de red y obtiene acceso a los datos cifrados, sin las claves criptográficas adecuadas no podrá leer ni utilizar esos datos. Los otros métodos eficaces para proteger los datos almacenados deberían considerarse oportunidades para mitigar el riesgo posible.” y requisito 4: “Codifique la transmisión de los datos de los titulares de tarjetas a través de redes públicas abiertas. La información confidencial se debe codificar durante su transmisión a través de redes a las que delincuentes puedan acceder fácilmente. Las redes inalámbricas mal configuradas y las vulnerabilidades en cifrados y protocolos de autenticación pueden ser los objetivos de delincuentes que explotan estas vulnerabilidades a los efectos de acceder a los entornos de datos de los titulares de las tarjetas.”

En esta sección se desarrollaran las técnicas que pueden ser utilizadas para mitigar el riesgo de fuga de información a través de la modificación de los datos antes de su pasaje a un ambiente de testing o desarrollo, estas técnicas son aplicadas exclusivamente a datos en reposo.

5.4.1 TDE

Una de las características disponibles en SQL Server 2014 es la denominada TDE (Transparent Data Encryption) de acuerdo a al artículo de Microsoft “Cifrado

de datos transparente" [20]: "El *Cifrado de datos transparente* (TDE) cifra los archivos de datos de SQL Server, base de datos, lo que se conoce como cifrado de datos en reposo."
"TDE realiza el cifrado y descifrado de E/S en tiempo real de los datos y los archivos de registro. El cifrado utiliza una clave de cifrado de la base de datos (DEK – Data Encryption Key), que está almacenada en el registro de arranque de la base de datos para que esté disponible durante la recuperación. La DEK es una clave simétrica protegida utilizando un certificado almacenado en la base de datos maestra del servidor o una clave asimétrica protegida por un módulo EKM. TDE protege los datos "en reposo", es decir, los archivos de datos y de registro. Ofrece la posibilidad de cumplimiento con leyes, normativas y directrices establecidas en diversos sectores."

Como se menciona en [20] el cifrado y descifrado de datos es transparente para el usuario. "El cifrado del archivo de base de datos se realiza en el nivel de página. Las páginas de una base de datos cifrada se cifran antes de escribirse en el disco y se descifran cuando se leen en la memoria. TDE no aumenta el tamaño de la base de datos cifrada."

Cuando se activa esta característica los datos en disco permanecen siempre cifrados así como también todos los backups realizados, con lo cual en caso de que se comprometan los archivos de la base de datos, los famosos mdf y ldf respectivamente (son conocidos así debido a la extensión del archivo que poseen) estos archivos no serán útiles ya que para que los mismos puedan ser descifrados es necesario contar con el certificado que protege la clave simétrica con la cual están cifrados dichos archivos.

5.4.2 Cifrado de Backups

Una nueva característica que se incorpora en SQL Server 2014 es la de tener la posibilidad de realizar backups cifrados. De acuerdo a la publicación en el sitio de Microsoft "Cifrado de copia de seguridad" [21]: "A partir de SQL Server 2014,

SQL Server tiene la capacidad de cifrar los datos mientras crea una copia de seguridad. Al especificar el algoritmo y el sistema de cifrado (un certificado o una clave asimétrica) al crear una copia de seguridad, puede crear un archivo de copia de seguridad cifrado"

Los tipos de cifrado y los sistemas soportados son según [21] son: "Para cifrar durante la copia de seguridad, debe especificar un algoritmo y un sistema cifrado para proteger la clave de cifrado. A continuación se exponen las opciones admitidas de cifrado:

- **Algoritmo de cifrado:** los algoritmos de cifrado admitidos son AES 128, AES 192, AES 256 y Triple DES.
- **Sistema de cifrado:** un certificado o una clave asimétrica.

Para realizar una restauración de la base de datos se requiere que el certificado o la clave simétrica que fue utilizada para cifrar el archivo de copia de seguridad estén disponibles en la instancia en la que se está realizando la restauración. La cuenta de usuario que realiza la restauración debe tener permisos de **VIEW DEFINITION** en el certificado o la clave. Si se restaura la copia de seguridad cifrada en una instancia diferente, debe asegurarse de que el certificado esté disponible en esa instancia.

La utilización de este tipo de copias de seguridad es una alternativa a la vista anteriormente (TDE), aunque podrían utilizarse las dos características de manera conjunta, teniendo algunas bases de datos con TDE y otras a las cuales se les aplica el cifrado durante la realización de la copia de seguridad.

Siempre que se utilicen en conjunto ambas estrategias para la realización de las copias de seguridad es recomendable que se manejen dos certificados digitales diferentes, para reducir el riesgo de fuga de información en caso de que alguno de los certificados sea comprometido.

5.4.3 Clasificación de la información

El proceso de clasificación de la información consiste en poder diferenciar cual es la información sensible que se encuentra dentro de la base de datos a proteger y de esta manera poder asignarle un nivel de seguridad adecuado. Esta definición

de niveles de sensibilidad deberá ser plasmada dentro de las políticas de seguridad para así poder definir que usuarios podrán acceder a la información y que usos estarán habilitados a darle a dicha información.

Una estrategia útil para realizar la protección de la información en base a la clasificación realizada es la que se menciona en [19]:

- “Definir algunas "clases de sensibilidad" sencillas que pueden usarse para agrupar columnas de datos según su nivel de sensibilidad.
- Crear roles de base de datos a través de los cuales podamos controlar el acceso a cada clase de datos.
- Asignar la membresía de cada rol
- Utilizar las propiedades extendidas de SQL Server para asignar una clase de sensibilidad a cada columna de la base de datos.”

Cabe destacar la utilización de las propiedades extendidas de SQL Server (último punto del esquema mencionado anteriormente), las mismas son muy útiles para poder identificar los objetos (columnas, tablas, etc.) a los cuales se les asignará un nivel de sensibilidad, como propone Microsoft en su artículo “Usar propiedades extendidas en objetos de base de datos” [22]“Al usar propiedades extendidas, puede agregar texto, por ejemplo contenido descriptivo o instructivo, agregar máscaras de entrada, y agregar reglas de formato como propiedades de objetos de una base de datos o de la base de datos misma. Por ejemplo, puede agregar una propiedad extendida a un esquema, a una vista de esquema o a una columna de la vista. Como las propiedades extendidas se almacenan en la base de datos, todas las aplicaciones que leen las propiedades pueden evaluar el objeto de la misma manera. Esto ayuda a exigir coherencia en la forma en que todos los programas tratan a los datos en el sistema.

Las propiedades extendidas se pueden utilizar para lo siguiente:

- Especificar un título para una tabla, vista o columna. De esta manera, las aplicaciones pueden utilizar el mismo título en una interfaz de usuario que muestre información de esa tabla, vista o columna.
- Especificar una máscara de entrada para una columna, de manera que las aplicaciones puedan validar datos antes de ejecutar una instrucción

Transact-SQL. Por ejemplo, el formato requerido para una columna de código postal o número de teléfono se puede especificar en la propiedad extendida.

- Especificar reglas de formato para mostrar los datos en una columna.
- Registrar una descripción de los objetos de base de datos específicos que las aplicaciones pueden mostrar a los usuarios. Por ejemplo, se pueden usar descripciones en una aplicación o un informe del diccionario de datos.
- Especificar el tamaño y la ubicación de la ventana en la que se mostrará una columna."

Luego de definir clases de sensibilidad podría también utilizarse el cifrado para proteger las columnas sensibles como se menciona en la sección "Cifrado a nivel de celda".

5.4.4 Cifrado a nivel de celda

En Sql Server existen un conjunto de funciones para realizar un cifrado simétrico a nivel de celda, estas funciones pueden utilizarse tanto para cifrar como para descifrar la información utilizando certificados digitales o claves asimétricas para proteger las claves simétricas que cifran y descifran las información, tal como se describe en "*Encrypt a column of data*"[23].

5.4.5 Sanitización de datos

Es muy común ver en las organizaciones que las bases de datos productivas se pasen (completas o en forma parcial) a los ambientes de prueba y/o desarrollo para que los testers y programadores puedan realizar pruebas con datos reales, esto se debe a que en muchos casos es difícil generar los datos para que las pruebas realizadas sobre los sistemas se asemejen los más posible al comportamiento que tendrán dichos sistemas en el ambiente productivo.

Aquí nos encontramos con un problema de seguridad, ya que los ambientes se testing y desarrollo en la mayoría de los casos no poseen el mismo nivel de seguridad que se aplica en los servidores de producción, con lo cual al pasar una base de datos productiva a un nuevo ambiente corremos el riesgo de estar exponiendo información sensible en ambientes no seguros.

También hay que tener en cuenta que en muchos países existen regulaciones relacionadas con la protección de los datos personales, lo que implica además un cumplimiento legal por parte de la organización.

Si bien es importante para los testers y los desarrolladores que la estructura de la base de datos con la cual van a trabajar sea correcta, no pasa lo mismo con los datos que contiene, aunque los datos no sean reales es suficiente con que parezcan reales, por eso la modificación de los datos antes del pasaje de ambiente se considera una buena práctica para la protección de dichos datos.

Para realizar la modificación de la información que se va trasladar al ambiente de pruebas existen varias técnicas que son conocidas como técnicas de sanitización de datos.

Estas son algunas de las técnicas utilizadas para realizar la sanitización de los datos según “Data Sanitization Techniques” [24]:

5.4.5.1 Valores nulos

Esta técnica consiste simplemente en borrar los datos de una o varias columnas, aunque es eficaz a la hora de evitar que se filtre información confidencial en ambientes de prueba, esta técnica no es muy utilizada ya que para realizar pruebas muchas veces se necesitan datos realistas.

5.4.5.2 Enmascaramiento de datos

Enmascarar datos significa sustituir un campo o parte del mismo con una máscara, esto permite mantener el formato del dato para ser mostrado por pantalla o en informes.

Un típico caso es el de enmascarar los números de tarjetas de crédito. La norma PCI-DSS [6] en su requisito 3.3 exige que los números de tarjeta se encuentren enmascarados, tanto a la hora de exhibirlos como durante su almacenamiento (requisito 3.4).

5.4.5.3 Sustitución

Esta técnica consiste en reemplazar el contenido de una columna de datos con información que es similar a los datos reales. Por ejemplo, los apellidos en una base de datos de clientes podrían ser modificados mediante la sustitución del valor real con apellidos extraídos de una lista previamente confeccionada.

5.4.5.4 Mezclado

El mezclado de datos es similar a la sustitución, excepto que la sustitución de datos se realiza con datos de la propia columna. Básicamente los datos se mueven al azar o con algún algoritmo definido entre las filas hasta que ya no hay ninguna correlación razonable con la información restante en la fila. Esta técnica es útil para casos en los cuales los datos tienen algún tipo de validación y no pueden ser inventados por características inherentes a los mismos, por ejemplo los números de cuil.

5.4.5.5 Varianza numérica

Esta técnica implica la modificación de cada valor numérico en una columna por un porcentaje aleatorio de su valor real, este tipo de cambio es bastante bueno para mantener una distribución dentro de rangos aceptables para la información que se está manejando. Por ejemplo incrementar un porcentaje variable sobre una determinada columna.

5.4.6 Arquitectura

Otro enfoque posible en la protección de los datos es aquel que se basa en la estructura física donde se encuentran alojados los datos. Según [19] es recomendable aplicar una estrategia a nivel de arquitectura para proteger los datos: “El almacenamiento estratégico de datos y la abstracción de la organización subyacente de la base de datos proporcionan una manera de reducir el riesgo de divulgación completa de datos sensibles. Aumentar la cantidad de conocimientos necesarios requeridos para revelar datos sensibles fuera de los métodos establecidos reduce los jugadores involucrados en el campo de batalla.”

A continuación se nombran algunas técnicas de protección propuestas por John Magnabosco en [19]:

“La normalización ofrece una manera eficiente de almacenar datos dentro de una base de datos relacional. Para asegurar la normalización ofrece la separación de los datos sensibles de los datos que se considera menos sensible. Esta separación aumenta el nivel de revelación que tiene que ocurrir para hacer que los datos sensibles sean útiles para el ladrón de datos.

SQL Server ofrece una función de servidor vinculado (*Linked Servers*) que presenta la oportunidad de ampliar los beneficios de la normalización a través de múltiples servidores físicos; aumentando así los requisitos para obtener acceso a los datos sensibles que se almacenan en el servidor de base de datos diferente. Vistas y esquemas de base de datos son características a nivel de base de datos que ofrecen capas de abstracción del esquema subyacente de la base de datos que proporcionan una forma de proteger datos confidenciales y gestionar de forma más eficaz el acceso a ella.”

6 Auditoria en SQL Server 2014

SQL Server provee de varias características relacionadas con el registro de eventos en el servidor de base de datos. La recolección de estos eventos es útil para cumplir con normativas vigentes o en los casos en los cuales se requiera hacer un análisis forense sobre alguna situación producida en el servidor, como ser un acceso indebido o alguna fuga de información. Pero también esta información es importante para avisarnos en tiempo real sobre algún evento en particular que hayamos definido, como por ejemplo inicios de sesión fallidos de un usuario administrador, esto podría indicarnos que se está produciendo un ataque sobre la cuenta mencionada y nos daría la posibilidad de reaccionar y analizar los eventos que se produjeron.

6.1 Configuración de archivos de registros (logs)

En el registro de errores de SQL Server se puede encontrar información sobre lo que está ocurriendo en el servidor de base de datos. Cada registro de errores de SQL Server tiene toda la información relacionada con los fallos y/o errores que han ocurrido desde que el servidor de SQL fue reiniciado por última vez o desde la última vez que se han reciclado los registros de errores. De forma predeterminada se almacenan seis archivos de registros de errores de SQL Server más un séptimo archivo que es el que se encuentra activo, cuando este archivo se llena se elimina el más antiguo de los 6 archivados. A pesar de que 6 archivos parecen suficientes, es una buena práctica aumentar el número de registros de errores de SQL Server, hay que tener en cuenta que cada vez que se reinicia una instancia de SQL Server un nuevo archivo de registro es creado como se explica en “Configure SQL Server Error Logs” [25]. Es importante que se realice una copia de resguardo de estos archivos antes de ser eliminados.

6.2 Auditoría de logins

Dentro de SQL Server existen cuatro opciones de configuración para registrar los intentos de login, como se menciona en el artículo de Technet “Server Properties (Security Page)” [26], las mismas son:

- intentos fallidos de login,
- intentos exitosos de login,
- intentos exitosos y fallidos de login y
- ninguno

Aquí la recomendación es establecer la opción de intentos de login fallidos (opción por defecto en SQL Server 2014), dado que este tipo de información podría ayudar a detectar algún tipo de ataque de adivinación de contraseña. Si bien es útil contar con la información de los logins exitosos, los mismos pueden registrarse por otros medios y evitar de este modo llenar los archivos de registros de errores.

La habilitación de esta opción se hace a nivel de instancia y desde la sección “Seguridad” en el SQL Server Management Studio.

También se puede consultar el valor establecido con la ejecución de la siguiente sentencia:

```
XP_loginconfig 'audit level';
```

La modificación de esta característica requiere el reinicio del servicio de SQL Server.

6.3 SQL Server Audit

SQL Server incluye SQL Server Audit, la cual es una funcionalidad que permite registrar eventos que suceden tanto a nivel de servidor como a nivel de base de datos. Dicha funcionalidad permite mayor granularidad al momento de definir que eventos deben ser registrados.

Como define Microsoft en su artículo “SQL Server Audit” [27] “Puede utilizar SQL Server Management Studio o Transact-SQL para definir una auditoría. Una vez creada y habilitada la auditoría, el destino comenzará a recibir entradas.
Puede leer los registros de eventos de Windows mediante la utilidad **Visor de eventos** en Windows. Para los destinos de archivo, puede utilizar tanto el **Visor del archivo de registros** en SQL Server Management Studio como la función fn_get_audit_file para leer el archivo de destino.
El proceso general de creación y uso de una auditoría es el siguiente:

- Cree una auditoría y defina el destino.
- Cree una especificación de auditoría de servidor o una especificación de auditoría de base de datos que se asigne a la auditoría. Habilite la especificación de auditoría.
- Habilite la auditoría.
- Lea los eventos de auditoría mediante el Visor de eventos o el Visor de archivos de registro de Windows, o la función fn_get_audit_file.”

Microsoft define como buenas prácticas para SQL Audit escribir los eventos de en el registro de seguridad de Windows (Escribir eventos de auditoría de SQL Server en el registro de seguridad) [28]: “En un entorno de alta seguridad, el registro de seguridad de Windows es la ubicación adecuada para escribir los eventos que registran el acceso a los objetos. Se admiten otras ubicaciones de auditoría pero están más expuestas a alteraciones.
La directiva de auditoría de Windows puede afectar a la auditoría de SQL Server si se configura para escribir en el registro de seguridad de Windows, con la posibilidad de perder eventos si la directiva de auditoría se configura incorrectamente. Por lo general, el registro de seguridad de Windows se establece para sobrescribir los eventos más antiguos. De esta forma se conservan los eventos más recientes. Sin embargo, si el registro de seguridad de Windows no se establece para sobrescribir los eventos más antiguos, entonces, si el registro de seguridad se llena, el sistema emitirá el evento 1104 de Windows (el registro está lleno). En ese punto:

- No se registrará ningún evento de seguridad más

- SQL Server no podrá detectar que el sistema no puede grabar eventos en el registro de seguridad, lo que provoca la posible pérdida de eventos de auditoría
- Después de que el administrador corrija el registro de seguridad, el comportamiento de registro volverá a la normalidad."

6.4 SQL Trace

"SQL Trace es una tecnología ligera, pero potente, que puede ejecutarse en SQL Server; recopila datos de rendimiento seleccionados de cientos de puntos de datos de rendimiento posibles que van desde bloqueos, conexiones, instrucciones SQL DML y recompilación" según lo define Adam Jorgensen en Microsoft SQL Server 2012 Bilble [29].
"Microsoft SQL Server ofrece procedimientos almacenados del sistema Transact-SQL para crear seguimientos en una instancia del Motor de base de datos de SQL Server. Puede utilizar estos procedimientos almacenados del sistema desde sus propias aplicaciones para crear seguimientos manualmente, en lugar de utilizar el SQL Server Profiler. Esto permite escribir aplicaciones personalizadas específicas para las necesidades de la organización", según se menciona en el artículo de Microsoft Docs "SQL Trace" [30].

Este tipo de recopilación de información puede utilizarse como una alternativa a la opción antes vista de SQL Server Audit [27] cuando no se posee una licencia de SQL Enterprise ya que se requiere esta versión para auditar determinados eventos a nivel de base de datos y no de servidor.
Con SQL Trace es posible crear distintos "trace" y configurarlos para capturar diferentes eventos tanto a nivel de servidor como a nivel de base de datos. Los resultados pueden almacenarse tanto en tablas de una base de datos como en archivos circulares que se van sobre escribiendo a medida que llegan a un tamaño preestablecido.

Cabe destacar que SQL Trace será discontinuado en versiones futuras de acuerdo a [30] “Esta característica se quitará en una versión futura de Microsoft SQL Server. Evite utilizar esta característica en nuevos trabajos de desarrollo y tenga previsto modificar las aplicaciones que actualmente la utilizan. Use eventos extendidos en su lugar.”

6.5 Logon Triggers

En SQL Server existen unos tipos especiales de triggers denominados triggers de logon, estos pueden utilizarse para restringir las aplicaciones que se utilizan para conectarse al servidor, restringir las conexiones dependiendo del host del cual provienen, evitar que usuarios se conecten en determinados horarios. De acuerdo con Microsoft en su artículo “Desencadenadores Logon” [31] “Los desencadenadores LOGON activan procedimientos almacenados en respuesta a un evento LOGON. Este evento se genera cuando se establece una sesión de usuario con una instancia de SQL Server. Los desencadenadores logon se activan después de que termine la fase de autenticación del inicio de sesión, pero antes de que se establezca la sesión de usuario realmente. Por tanto, todos los mensajes que se originan dentro del desencadenador y que normalmente llegarían hasta el usuario, como los mensajes de error y los mensajes de la instrucción PRINT, se desvían al registro de errores de SQL Server. Los desencadenadores logon no se activan si se produce un error en la autenticación. Puede utilizar desencadenadores logon para realizar auditorías y controlar sesiones de servidor, como el seguimiento de la actividad de inicio de sesión, la restricción de inicios de sesión en SQL Server o la limitación del número de sesiones para un inicio de sesión específico.”

7 Configuración del firewall de Windows

Una adecuada configuración del firewall de Windows deberá ser uno de los primeros pasos al instalar el SQL Server para que los puertos necesarios para realizar la conexión a dicho motor no se encuentren accesibles desde fuera del servidor. De acuerdo con el artículo de Micorosoft "Configurar el Firewall de Windows para permitir el acceso a SQL Server" [32]: "Para tener acceso a una instancia de SQL Server a través de un firewall, debe configurar el servidor de seguridad en el equipo que ejecuta SQL Server para permitir el acceso. El firewall es un componente de Microsoft Windows. También puede instalar un firewall de otra empresa. Este tópico describe cómo configurar el servidor de seguridad de Windows, pero los principios básicos se aplican a otros programas de cortafuegos."

La siguiente tabla [32] tiene listados los puertos en los cuales se encuentran los servicios de SQL Server:

Scenario	***Port***	***Comments***
SQL Server default instance running over TCP	*TCP port 1433*	*This is the most common port allowed through the firewall. It applies to routine connections to the default installation of the Database Engine, or a named instance that is the only instance running on the computer. (Named instances have special considerations. See Dynamic Ports later in this topic.)*
SQL Server named instances in the default configuration	*The TCP port is a dynamic port determined at the time the Database Engine starts.*	*See the discussion below in the section Dynamic Ports. UDP port 1434 might be required for the SQL Server Browser Service when you are using named instances.*
SQL Server named instances when they are configured to use a fixed port	*The port number configured by the administrator.*	*See the discussion below in the section Dynamic Ports.*

Dedicated Admin Connection	*TCP port 1434 for the default instance. Other ports are used for named instances. Check the error log for the port number.*	*By default, remote connections to the Dedicated Administrator Connection (DAC) are not enabled. To enable remote DAC, use the Surface Area Configuration facet. For more information, see Surface Area Configuration.*
SQL Server Browser service	*UDP port 1434*	*The SQL Server Browser service listens for incoming connections to a named instance and provides the client the TCP port number that corresponds to that named instance. Normally the SQL Server Browser service is started whenever named instances of the Database Engine are used. The SQL Server Browser service does not have to be started if the client is configured to connect to the specific port of the named instance.*
SQL Server instance running over an HTTP endpoint.	*Can be specified when an HTTP endpoint is created. The default is TCP port 80 for CLEAR_PORT traffic and 443 for SSL_PORT traffic.*	*Used for an HTTP connection through a URL.*
SQL Server default instance running over an HTTPS endpoint.	*TCP port 443*	*Used for an HTTPS connection through a URL. HTTPS is an HTTP connection that uses secure sockets layer (SSL).*
Service Broker	*TCP port 4022. To verify the port used, execute the following query:* `SELECT name, protocol_desc, port, state_desc FROM sys.tcp_endpoints WHERE type_desc = 'SERVICE_BROKER'`	*There is no default port for SQL ServerService Broker, but this is the conventional configuration used in Books Online examples.*

Database Mirroring	*Administrator chosen port. To determine the port, execute the following query:* `SELECT name, protocol_desc, port, state_desc FROM sys.tcp_endpoints WHERE type_desc = 'DATABASE_MIRRORING'`	*There is no default port for database mirroring however Books Online examples use TCP port 7022. It is very important to avoid interrupting an in-use mirroring endpoint, especially in high-safety mode with automatic failover. Your firewall configuration must avoid breaking quorum. For more information, see Specify a Server Network Address (Database Mirroring).*
Replication	*Replication connections to SQL Server use the typical regular Database Engine ports (TCP port 1433 for the default instance, etc.)*	*For sync over HTTP, replication uses the IIS endpoint (ports for which are configurable but is port 80 by default), but the IIS process connects to the backend SQL Server through the standard ports (1433 for the default instance.*
	Web synchronization and FTP/UNC access for replication snapshot require additional ports to be opened on the firewall. To transfer initial data and schema from one location to another, replication can use FTP (TCP port 21), or sync over HTTP (TCP port 80) or File Sharing. File sharing uses UDP port 137 and 138, and TCP port 139 if it using NetBIOS. File Sharing uses TCP port 445.	*During Web synchronization using FTP, the FTP transfer is between IIS and the SQL Server publisher, not between subscriber and IIS.*

Transact-SQL debugger	*TCP port 135* *See Special Considerations for Port 135* *The IPsec exception might also be required.*	*If using Visual Studio, on the Visual Studio host computer, you must also add* ***Devenv.exe*** *to the Exceptions list and open TCP port 135.* *If using Management Studio, on the Management Studio host computer, you must also add ssms.exe to the Exceptions list and open TCP port 135. For more information, see Configure the Transact-SQL Debugger.*

8 Conclusión

SQL Server brinda un amplio conjunto de características que implementado de forma correcta y planificada puede lograr una mejora notable en la seguridad de los datos. Además, permite una flexibilidad de adaptación que hace que sea posible definir un plan de implementación de sus características tanto para una empresa pequeña como para una gran empresa, cada una de ellas con sus respectivos requerimientos y necesidades particulares.

El nivel de seguridad implementado en cada instalación depende directamente del nivel de conocimiento de las características de seguridad por parte de los responsables de implementar la seguridad en el servidor. Y es por esto que es importante conocer cuáles son esas características y cómo se configuran correctamente, ya que de lo contrario se podría llegar a omitir o implementar de manera errónea dichas funcionalidades. En este trabajo se exploraron diversas funcionalidades las cuales fueron probadas y adecuadas tratando de utilizar para cada funcionalidad las características de seguridad disponibles.

A partir del análisis y las pruebas realizadas sobre el motor de base de datos SQL Server 2014 y el sistema operativo Windows Server 2012 R2 en cuanto a cuestiones de configuración de características de seguridad se pudo verificar que la aplicación de las mismas permitió transformar una instalación pobremente asegurada en una plataforma fuertemente protegida y todo esto fue posible sin tener que implementar herramientas de terceros, únicamente configurando correctamente el motor de base de datos y el sistema operativo.

Aunque algunas de las características mencionadas en este trabajo podrían llegar a ser difíciles de implementar en sistemas ya en funcionamiento (por ejemplo la sanitización de datos en diseños que no se encuentran normalizados se complica bastante ya que la redundancia de información implica un esfuerzo extra al reemplazar datos, o también la versión del motor instalada que impide aplicar ciertas características en versiones estándar de SQL Server 2014 como la auditoría fina a nivel de base de datos) la mayoría de las características de

seguridad son aplicables y en muchos casos con un bajo/nulo costo de implementación

9 Bibliografía

[1] Jerome H. Saltzer and M. Schroeder, “The Protection of Information in Computer Systems”, Massachusetts Institute of Technology, Cambridge, Octubre 1974.

[2] Federal Information Processing Standards, “Standards for Security Categorization of Federal Information and Information Systems”, NIST, 2004.

[3] Julia Allen *et al*, “Securing Network Servers”, Carnegie Mellon Software Engineering Institute, Pittsburgh, Abril 2000.

[4] Karen Scarfone *et al*, “Guide to General Server Security”, NIST, Gaithersburg, Julio 2008.

[5] Richard Kissel *et al*, “Guidelines for Media Sanitization”, NIST, Gaithersburg, Diciembre 2014.

[6] PCI Security Standards Council, PCI-DSS 3.2, PCI Security Standards Council, Abril 2016

[7] Microsoft, “Asistente para configuración de seguridad”, https://technet.microsoft.com/es-us/library/cc754997(v=ws.11).aspx , 2017

[8] Microsoft, “Microsoft Baseline Security Analyzer”, https://technet.microsoft.com/es-ar/security/cc184924.aspx , 2017

[9] Microsoft, “Microsoft Security Compliance Manager”, https://technet.microsoft.com/es-us/library/cc677002.aspx, 2017

[10] SomarSoft, “SomarSoft Utilities”, http://www.systemtools.com/somarsoft/index.html, 2017

[11] Technet Microsoft, “AccessEnum v1.32”, https://technet.microsoft.com/en-us/sysinternals/bb897332, 2017

[12] Microsoft, “Autenticación en SQL Server”, https://msdn.microsoft.com/es-es/library/bb669066(v=vs.110).aspx, 2017

[13] Microsoft, “Configurar los permisos y las cuentas de servicio de Windows”, https://**msdn**.microsoft.com/es-es/library/ms143504.aspx#VA_Desc, Octubre 2016

[14] TechNet Microsoft, “Introducing Managed Service Accounts”, https://technet.microsoft.com/en-us/library/dd560633(v=ws.10).aspx, 2011

[15] John Savill, “What’s a virtual account?”, http://windowsitpro.com/systems-management/q-whats-virtual-account, 2009

[16] Microsoft, “ALTER LOGIN (TRANSACT-SQL)”, https://msdn.microsoft.com/es-es/library/ms189828(v=sql.110).aspx, 2016

[17] Microsoft, “Politicas de seguridad de Windows”, Extraído del comando secpol.msc en Windows Server 2012R2, 2016

[18] Microsoft, “Directiva de contraseñas”, https://msdn.microsoft.com/es-es/library/ms161959(v=sql.120).aspx, 2016

[19] John Magnabosco, “Protecting SQL Server Data”, ISBN: 978-1-906434-26-7, 2009

[20] Microsoft, “Cifrado de datos transparente (TDE)”, https://msdn.microsoft.com/es-es/library/bb934049.aspx, Octubre 2016

[21] Microsoft, “Cifrado de copia de seguridad”, https://msdn.microsoft.com/es-es/library/dn449489.aspx, Octubre 2016

[22] Technet Microsoft, “Usar propiedades extendidas en objetos de base de datos”, https://technet.microsoft.com/es-ar/library/ms190243(v=sql.105).aspx, 2017

[23] Microsoft Docs, “Encrypt a column of data”, https://docs.microsoft.com/en-us/sql/relational-databases/security/encryption/encrypt-a-column-of-data, 2017

[24] Dale Edgar, “Data Sanitization Techniques”, http://www.orafaq.com/papers/data_sanitization.pdf, 2004

[25] Microsoft, “Configure SQL Server Error Logs”, https://msdn.microsoft.com/en-us/library/ms177285(v=sql.120).aspx, 2016

[26] Microsoft, “Server Properties (Security Page)”, https://technet.microsoft.com/en-us/library/ms188470(v=sql.120).aspx, 2016

[27] Microsoft, “SQL Server Audit (Motor de base de datos)”, https://msdn.microsoft.com/es-es/library/cc280386(v=sql.120).aspx, 2016

[28] Microsoft, “Escribir eventos de auditoría de SQL Server en el registro de seguridad”, https://msdn.microsoft.com/es-es/library/cc645889(v=sql.120).aspx , 2016

[29] Adam Jorgensen, “Microsoft SQL Server 2012 Bilble”, https://books.google.com.ar/books?id=MYr9NplN_W0C&pg=PA919#v=onepage&q&f=false, 2017

[30] Microsoft Docs, “SQL Trace”, https://docs.microsoft.com/es-es/sql/relational-databases/sql-trace/sql-trace, 2017

[31] MSDN, “Desencadenadores Logon”, https://msdn.microsoft.com/es-es/library/bb326598(v=sql.120).aspx, 2017

[32] Microsoft, “Configurar el Firewall de Windows para permitir el acceso a SQL Server”, https://msdn.microsoft.com/en-us/library/cc646023(v=sql.120).aspx, 2017