Database Main Threats Analisys Using MS SQL Server

M.J. García

***Abstract*— Imperva's, information security specialized company, security databases threats for 2013 identifies the main reasons allowing internal or external attackers grant access to database servers, getting the data stored there and "impact business operations in addition to financial loss or reputation damage", the report analyses and identifies that a database server carefully configurated may help to mitigate those threats.**

This document analyses the main threats caused by a default configuration, a configuration presented on installation moment, or an easy way when implementing on live systems, allow an attacker to violate the database server. For instance privilege abuse, weak of audit trail, excessive and unused privileges, storage media exposure, and exploitation of vulnerabilities and misconfigured databases. In order to develop the current document an MS SQL Server will be used, showing risks to which the system is exposed by the above threats to finally propose how to mitigate using best practices.

***Keywords*— Database, misconfigured database risks, database best practices, SQL injection, database configuration.**

INTRODUCCION

Imperva en su análisis de 2013 muestra las diez amenazas más frecuentes a las bases de datos, las cuales coinciden con las amenazas reportadas en el año 2010 demostrando así que estas no han variado, tan solo algunas se han vuelto más populares que otras durante este periodo de tiempo, como es el caso por ejemplo de la inyección SQL de la cual se habló por primera vez en 1998, y al 2013 se ubica en el lugar número 3 del informe, causando preocupación por su incidencia y relativa sencillez para explotación.

El desconocimiento tanto de las amenazas a la seguridad en bases de datos como de estrategias para mitigarlas, llamado por Imperva como experiencia limitada en seguridad y falta de educación, permite a atacantes tomar ventaja de los administradores de estas y lograr obtener la información almacenada. En el presente documento se tratarán las principales amenazas a la seguridad de las bases de datos haciendo uso de un motor MS SQL Server Versión 2012, para ejemplo tanto de las amenazas como de las configuraciones propuestas al sistema para mitigarlas.

I. USO EXCESIVO DE PRIVILEGIOS INNECESARIOS

Hace referencia a usuarios con permisos más allá de los necesarios para ejercer su trabajo, por ejemplo un usuario cuyos accesos deben ser para consulta y generación de reportes contables, que tienen acceso a la creación de usuarios. El origen de esta amenaza a la seguridad se da al momento de realizar la creación de los usuarios por parte de los encargados de la administración (administrador de bases de datos o DBA, "DatabaseAdministrator" por sus siglas en inglés)al poseer una política de control de accesos no definida o poco acorde a la compañía [1], [4].

Permite a usuarios con credenciales autorizadas conectarse a la base de datos y consultar y/o modificar información no relativa al rol desempeñado en la compañía. Se evitaría al utilizar el "principio de privilegios mínimo" [5] concediendo a cada usuario los accesos requeridos a los objetos de la base de datos, tablas, procedimientos, funciones, y a las operaciones que deba realizar, consulta, actualización o creación de nuevos registros. Para mitigar este riesgo en una base de datos MS SQL Server Versión 2012 se debe tener en cuenta lo siguiente:

- Clasificar los objetos de la base de datos, tablas, procedimientos almacenados, funciones y vistas haciendo uso de esquemas ("schemas")[6], por ejemplo: contabilidad, gerencia, recursos humanos; esto de acuerdo a la política interna definida para control de acceso.
- Hacer uso del manejo de la asignación de permisos mediante roles, en vez de asignarlos directamente al usuario [7].
- Emplear el "principio del privilegios mínimo" [4], [6] en la asignación de permisos.
- Es muy recomendable desactivar la cuenta invitado. Solamente necesita acceso a la base de datos MSDB [6].
- Hacer uso de las herramientas de auditoría de MS SQL Server, este se discutirá más adelante en este documento.
- Monitorizar aquellos usuarios con permisos DBO, son los usuarios con permisos de administración en la base de datos

.

II. ABUSO DE PRIVILEGIOS

Esta amenaza es materializada cuando un usuario interno, con credenciales autorizadas y permisos definidos para la ejecución de sus labores en la compañía, hace uso de estos para beneficio personal y/o de terceros ajenos a la empresa [8]. En la ilustración 1, se aprecia una comparación presentada por Verizon [3], respecto a riesgos generados por empleados malintencionados.

M.J.García, Universidad Pontificia Bolivariana (UPB), Bucaramanga, Santander, Colombia, miguel.jose.garcia@hotmail.com

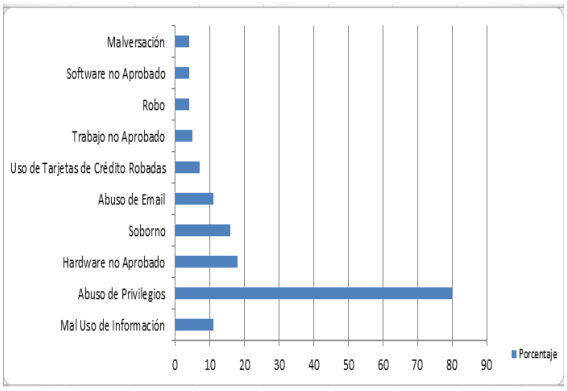

Ilustración 1. Riesgos generados por mal uso de información al interior de la compañía. Fuente: [3].

Realizando la revisión de la ilustración 1, muestra que entre las amenazas listadas, el abuso de privilegios, con un 80% de ocurrencia, es el riesgo al cual en mayor grado se exponen las compañías; convirtiéndose en uno de los mayores peligros enfrentados por estas [8]. La detección puede realizarse mediante el establecimiento de controles al interior de la base de datos como son:

- Realizar un seguimiento a las consultas ejecutadas. Como se puede observar en la ilustración 2, antes de realizar la operación de consulta, se realiza la creación de un registro de la misma en una tabla llamada Auditoría.

```
CREATE PROCEDURE [Contabilidad].[PA_TotalMovimientoDetallado]
    @Usuario        NVarchar(50),
    @IdMTPeriodo    Int,
    @Año            Int
AS
BEGIN
    SET NOCOUNT ON
    DECLARE @IdPeriodo   Int

    INSERT INTO [Control].[Auditoria] (Usuario, Fecha, Procedimiento, Operacion)
        VALUES (@Usuario, GETDATE(), 'Contabilidad.PA_TotalMovimientoDetallado', 'Consulta')

    SELECT @IdPeriodo  = IdMTPeriodo FROM MTPeriodo WHERE Periodo = @IdMTPeriodo  AND YEAR(FechaInicio) = @Año

    SELECT  M.IdMTPeriodo,
            ltrim(rtrim(CCostos.Descripcion)) + '-' + ltrim(rtrim(Cuentas.Descripcion)) AS 'Cuenta',
            M.Descripcion, M.Monto, M.TipoMovimiento
    FROM FN_MVComercioValorXMovimiento(@Año) M
        INNER JOIN CuentasXCCosto ON
            CuentasXCCosto.IdCuentasXCCosto = M.IdCuentasXCCosto AND CuentasXCCosto.EsActivo = 'True'
        INNER JOIN CCostos ON
            CCostos.IdCCosto = CuentasXCCosto.IdCCosto AND CCostos.EsActivo = 'True'
        INNER JOIN Cuentas ON
            Cuentas.idCuenta = CuentasXCCosto.IdCuenta AND Cuentas.EsActivo = 'True'
        WHERE M.IdMTPeriodo = @IdPeriodo
        ORDER BY Cuenta, M.TipoMovimiento
END
```

Ilustración 2. Ejemplo de seguimiento a la ejecución de consultas. Fuente Autor.

- Realizar una revisión de los planes de auditoría configurados a la base de datos en busca de patrones de conexión anómalos, por ejemplo conexiones, consultas fuera de horarios laborales y actualización o eliminación de registros [5].
- Registrar, de ser posible, intentos de conexión y/o conexiones realizadas empleando herramientas diferentes a las aplicaciones diseñadas para interactuar con la base de datos, por ejemplo SQLCMD [9] o herramientas de hoja de cálculo como MS Excel que permiten conectarse y consultar datos de MS SQL Server [10].

III. INYECCIÓN SQL

Se le define como una consulta válida a los requerimientos del negocio que en un momento determinado y con aplicación de conocimiento en lenguaje Transact-SQL, lenguaje de programación utilizado en MS SQL Server [11], es transformada para permitir a un atacante consultar y obtener información sensible. Un ejemplo sencillo de esta amenaza es el presentado en [2] en el año 1998 con la versión 6.5 de MS SQL Server vigente: "Select * From*Tabla1*Select * From*Tabla2*" y como se puede observar en la ilustración 3, en la cual se ejecuta un ejemplo de esta consulta en la versión MS SQL Server 2012 aún es posible realizarla.

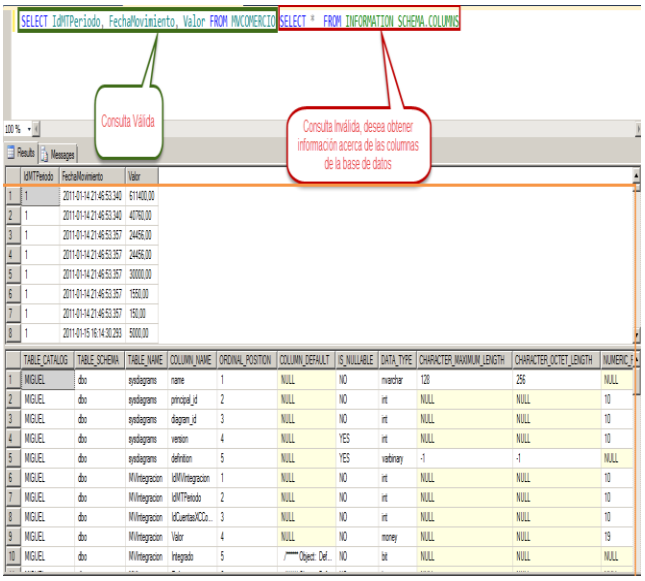

Ilustración 3. Ejemplo de inyección SQL. Fuente Autor.

Como se puede ver en la ilustración 3, la consulta en verde representa la consulta creada para satisfacer un requerimiento de la compañía mientras la consulta en rojo sería aquella creada por un atacante para obtener información de la base de datos, el resultado se enmarca en color naranja.

Las medidas adoptadas para mitigar este riesgo, son:

- Convertir a las consultas, procedimientos y funciones en parte activa de la estrategia de defensa [5], esto es no solo utilizar el lenguaje Transact-SQL para realizar las consultas si no emplearlo de manera tal que antes de ser ejecutada la consulta por el motor de base de datos, se realice un análisis de los parámetros enviados por el usuario desde la aplicación y así descartar tanto debilidades en el código como los caracteres especiales: "+","-","*","/", los símbolos empleados para realizar comentarios en el código: "--","/**/", las palabras reservadas empleadas por Transact-SQL, como select, insert, revoke, los caracteres usados para concatenar consultas como el ";" o " " entre consultas y así descartar un intento de concatenación [4] como el observado en la ilustración 1
- Realizar análisis de vulnerabilidades al código de ejecución de consultas a la base de datos.
- Procurar el uso de procedimientos almacenados, su uso dificulta la ejecución de consultas concatenadas. Además

que permiten al administrador una asignación de permisos a los usuarios para realizar tareas específicas en la base de datos sin tener que otorgar permisos sobre el grupo de objetos involucrados en esta [12].

Restringir los accesos y privilegios de los usuarios a los niveles mínimos [5], esto evitaría que puedan realizar consultas a las bases de datos del sistema mediante esta técnica

IV. MALWARE

Son aplicaciones diseñadas con un fin particular, desde mostrar un mensaje en el computador de un usuario hasta tomar control de un sistema encargado de una instalación SCADA, el caso de Stuxnet en el cual las instalaciones para investigación y desarrollo de la energía nuclear de una nación, Irán, fueron infectadas con este malware el cual fue desarrollado con la capacidad de controlar las señales eléctricas de un sistema tan complejo como delicado, las centrifugadoras utilizadas en el proceso, para hacer que estas trabajaran en condiciones límite y le mostraran al operador humano señales de operación normal, consiguiendo de esta manera sabotear las instalaciones nucleares Iraníes [13]. El uso de este tipo de herramientas se enmarca dentro de ataques más elaborados [1], ataques que pretenden tomar el control de los equipos de manera silenciosa ante los administradores. Su estrategia de propagación [14] hace uso de métodos como mensajes al correo electrónico, enlaces a sitios web, software pirata, ingeniería social y memorias USB; medio utilizado para propagar a Stuxnet, infectando tanto los dispositivos móviles como los dispositivos de almacenamiento portable (por ejemplo memorias USB) del personal con acceso autorizado, por ejemplo los contratistas [13].

Algunas medidas que pueden ser adoptadas para protegerse contra este tipo de amenazas, mitigando el riesgo de infección son:

- Evitar el uso de software pirata tanto en los equipos de los usuarios como en servidores.
- Ser cuidados con el tipo de páginas visitadas y las descargas realizadas.
- Mantener tanto la el motor de base de datos como el servidor donde ha sido instalada actualizado al día.
- Hacer uso de herramientas firewall, tanto físicas como lógicas [15].
- Restringir el acceso de la base de datos a la red externa, Internet, permitiendo solo acceso desde y hacia la red interna de la compañía, evitando conexiones indeseadas [15].
- Evitar otorgar permisos de súper usuario o administrador de base de datos (rol DBO) a los usuarios [5].

IV. FALTA DE AUDITORÍA

Es una tarea de vital importancia, la cual no debe evitar realizarse bajo pretextos de degradación de rendimiento y/o desempeño del motor de bases de datos. Permite a los administradores conocer las actividades realizadas por los usuarios y las consultas ejecutadas en el motor de base de datos, para así generar controles deseguridad eficaces y confiables [5].

MS SQL Server 2012, cuenta con algunas características que permiten a los administradores tomar ventaja de estas tareas [16]:

- Crear múltiples tareas de auditoría.
- Auditar la ejecución de actividades en ves las trazas.
- La auditoría reinicia con la instancia del motor de base de datos
- Las tareas de auditoría pueden ser detenidas por usuario con permisos suficientes más sin embargo, se registra quien lo hizo.

La ilustración 4 muestra un ejemplo de cómo realizar la creación de una tarea de auditoría mediante el uso de código Transact-SQL.

```
CREATE DATABASE BDPrueba;
GO
USE BDPrueba;
GO
CREATE SCHEMA SchemaInfo;
GO
CREATE TABLE SchemaInfo.DataGeneral (ID int PRIMARY KEY, Info varchar(50) NOT NULL);
GO
CREATE TABLE SchemaInfo.DataConfidencial (ID int PRIMARY KEY, Info varchar(50) NOT NULL);
GO
USE master;
GO
CREATE SERVER AUDIT AuditAccesoInfo
    TO FILE ( FILEPATH ='D:\SQLAuditoria\' )
    WHERE object_name = 'DataConfidencial' ;
GO
ALTER SERVER AUDIT AuditAccesoInfo WITH (STATE = ON);
GO
USE BDPrueba;
GO
CREATE DATABASE AUDIT SPECIFICATION [FiltroInformacionConfidencial]
FOR SERVER AUDIT [AuditAccesoInfo]
ADD (SELECT ON SCHEMA::[SchemaInfo] BY [public])
WITH (STATE = ON);
GO
SELECT ID, Info FROM SchemaInfo.DataGeneral;
SELECT ID, Info FROM SchemaInfo.DataConfidencial;
GO
SELECT * FROM fn_get_audit_file('D:\SQLAuditoria\AuditAccesoInfo_*.sqlaudit',default,default);
GO
```

Ilustración 4. Creación Plan de Auditoría. Fuente: Autor.

Como se puede apreciar resaltado en rojo, el código necesario para la creación del plan de auditoría con un filtro para definir que se desea registrar cualquier intento de consulta, operación Select, en la tabla DataConfidencial perteneciente al esquema "SchemaInfo" finalmente se realiza una consulta al plan de auditoría, almacenado en un archivo en disco duro. Finalmente en verde se observe la instrucción Transact-SQL requerida para realizar la lectura del archivo generado.

Como recomendaciones:

- Al momento previo a la activación de un plan de auditoría, es necesario realizar una revisión de los eventos que deseen auditarse, en caso contrario se podría estar generando una sobrecarga negativa en el servidor, impactando el desempeño de las bases de datos y generando un exceso de información innecesaria acerca de eventos y operaciones normales [5], [6].
- Revisar de manera periódica los resultados de la auditoría.

VI. EXPOSICIÓN DE LOS MEDIOS DE ALMACENAMIENTO

En este apartado se revisará el tema de la exposición de la información en los medios de copia de seguridad o back up; la ejecución de esta tarea es relativamente sencilla, escoger la hora indicada, el tipo de copia a realizar, incremental, diferencial o completa acto seguido el lugar en donde se almacenará y posteriormente decidir si será llevada a un medio externo, una cinta o un disco DVD pero, ¿qué sucede si esta copia es hurtada?, ¿Está protegida de manera que no permita una fácil lectura a quien la hurta? Como se puede observar en las ilustraciones 5 y6 una copia de seguridad creada por MS SQL Server en cualquier versión, para el presente apartado se tienen en cuenta desde la versión 2005, se escribe en texto claro lo cual permite leerla al abrirla con un editor de texto como el block de notas; cabe aclarar que el hecho de comprimirla no implica que se le cifre, solo se codifica de otra manera.

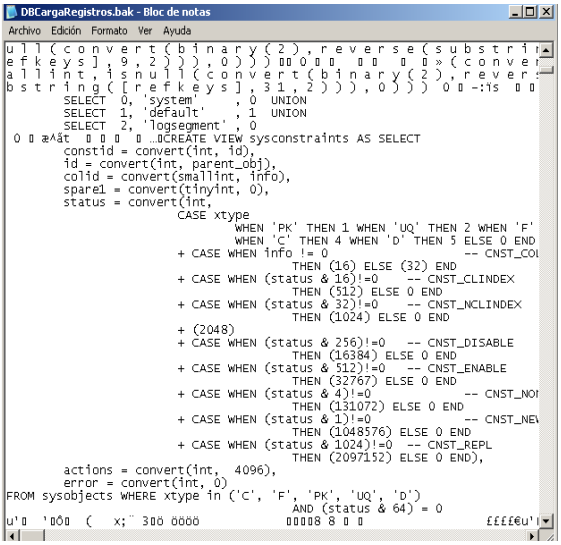

Ilustración 5. Copia de seguridad SQL Server 2005 sin comprimir o cifrar. Fuente: Autor.

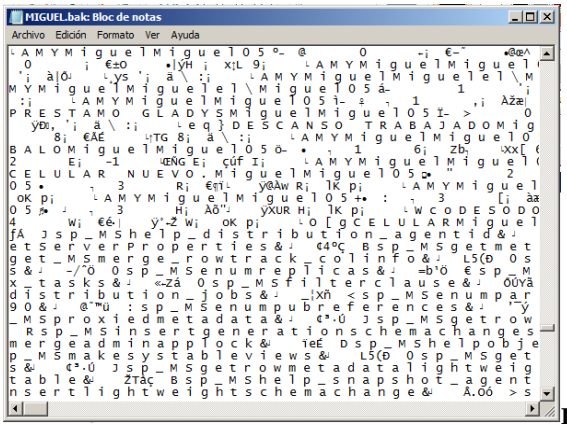

Ilustración 6. Copia de seguridad en MS SQL Server 2012 sin comprimir o cifrar. Fuente: Autor

Para evitar lo anteriormente expuesto hasta la versión 2012 de MS SQL Server se hace necesario realizar el cifrado de la base de datos utilizando una característica llamada TDE, Transparent Data Encription, [17], lo cual garantiza que al momento de realizar la copia de seguridad esta será cifrada por el motor y la restauración sin el certificado correcto, no será posible. En la ilustración 7 se observa el código Transact-SQL necesario para hacer uso de TDE.

```
USE master;
GO
CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'qwerty123+';
go
CREATE CERTIFICATE MyServerCert WITH SUBJECT = 'Certificado BDPrueba';
go
USE BDPrueba;
GO
CREATE DATABASE ENCRYPTION KEY
WITH ALGORITHM = AES_128
ENCRYPTION BY SERVER CERTIFICATE MyServerCert;
GO
ALTER DATABASE BDPrueba
SET ENCRYPTION ON;
GO
```

Ilustración 7. Activación de TDE. Fuente: Autor.

Resaltado en color naranja, se aprecia el código para crear la llave maestra para cifrado y el certificado a usar. En verde se observa cómo se activa TDE en la base de datos. Al activar TDE, se debe considerar el realizar una copia de seguridad tanto de las claves de cifrado como del certificado generado. En caso de necesitar realizar la restauración de la copia de seguridad en otro servidor y no contar con el certificado y/o la clave de cifrado empleada no sería posible realizar esta operación, generándose perdida de datos [17].

VII. EXPLOTACIÓN DE VULNERABILIDADES, BASES DE DATOS NO CONFIGURADAS

Motores de bases de datos sin actualizar, sistemas operativos sin actualizar, instalaciones de MS SQL Server utilizando las cuentas del sistema operativo (Local System, Network Service y Local Service) [18],bases de datos de producción restauradas en entornos para prueba sin medidas de protección adecuadas[5], [17], servidores web instalados en el mismo equipo en el que se encuentra el servidor de bases de datos, servidores de base de datos con acceso a y desde internet, bases de datos cuyos usuarios poseen excesivos privilegios[5]. Este listado de vulnerabilidades, encontrados en el desarrollo del presente documento, permitiría a atacantes, internos o externos, tomar control de la base de datos generando no solo pérdidas a la compañía si no también daño en su normal operación [1].

La manera de mitigar un ataque que se valga del uso de las vulnerabilidades tanto en la base de datos como del servidor donde se encuentra instalada son:

- Diseñar un plan de actualizaciones que permita mantener tanto el sistema operativo como el motor de base de datos actualizado; este plan debe incluir la ejecución de pruebas antes de aplicar las actualizaciones en los servidores de producción [5]
- Evitar utilizar las cuentas del sistema operativo (Local System, Network Service y Local Service) como las cuentas de inicio de sesión para los servicios del motor de base de datos. Estas cuentas están reservadas para la operación propia de los servicios y componentes del núcleo de Windows y como tal son cuentas con un alto nivel de privilegios sobre el sistema operativo. La cuenta Local System, Sistema local es utilizada por el sistema operativo para las operaciones del núcleo de Windows, esta cuenta es la encargada de ejecutar los procesos del sistema, el proceso de autenticación de usuarios con la aplicación%SystemRoot%\System32\
  Winlogon.exe, el administrador de sesión con laaplicación:%SystemRoot%\
  System32\Smss.exe y los procesos del subsistema: Csrss.exe además de poseer acceso completo sobre el registro de Windows y casi cualquier archivo almacenado, propio del sistema operativo o del usuario. La cuenta Network Service, Servicio de red, permite autenticar el equipo con otros equipos en una red Windows. La cuenta Local Service, Servicio local, posee privilegios similares a la cuenta Network Service con la diferencia que su acceso a recursos compartidos se limita a aquellos que permiten acceso anónimo [19].
- Emplear cuentas de inicio de sesión pertenecientes al dominio y con un nivel de privilegios mínimo [18].
- Evitar configurar el servidor web en el mismo servidor donde se encuentra el servidor de bases de datos, si el servidor web es comprometido por un atacante la información almacenada en la base de datos será hurtada con mayor facilidad [15].
- Realizar el cifrado del canal de comunicación con el servidor de base de datos.
- Evitar el acceso al servidor de base de datos desde la red externa, incluso la comunicación directa con este en la red interna. Utilizar reglas de firewall que puedan bloquear la comunicación indebida con este.
- Evitar la asignación de permisos innecesarios a los usuarios.
- Identificar los eventos que se consideran potencialmente peligrosos a la compañía con respecto al servidor de base de datos y monitorizarlos.

VIII. INFORMACIÓN SENSIBLE SIN ADMINISTRAR

Como se debe conocer el tipo de información almacenado en las tablas de las bases de datos, se hace necesario realizar y mantener un inventario detallado de las mismas, tipos de datos, nombres de las columnas, usuarios con acceso y nivel de permisos. Inclusive a las bases de datos en pruebas [1], [5]. Para este propósito MS SQL Server cuenta con algunas vistas y tablas del sistema que con el conocimiento y experticia en Transact-SQL pueden ser usadas para conocer las bases de datos alojadas, los permisos a las mismas y las tablas y tipos de datos de cada columna. En la ilustración 8 se pueden observar algunos ejemplos de estas consultas.

```
--Consultar las tablas y columnas
SELECT SO.NAME, SC.NAME FROM sys.objects SO
    INNER JOIN sys.columns SC ON SO.OBJECT_ID = SC.OBJECT_ID
WHERE SO.TYPE = 'U' ORDER BY SO.NAME, SC.NAME
--Consultar las columnas de una tabla
Select  * from information_schema.columns WHERE TABLE_NAME='DataConfidencial'
--Consultar los permisos a una base de datos
SELECT DB_NAME(st.dbid) AS database_name,
    OBJECT_SCHEMA_NAME(st.objectid, st.dbid) AS schema_name,
    OBJECT_NAME(st.objectid, st.dbid) AS object_name,
    st.text AS query_text
FROM sys.dm_exec_query_stats AS qs
CROSS APPLY sys.dm_exec_sql_text(qs.sql_handle) AS st
WHERE st.objectid IS NOT NULL AND ST.dbid = 4;
```

Ilustración 8. Ejemplos consultas para conocer permisos, tipos de datos y columnas de Bases de datos. Fuente: Autor.

IX. DENEGACIÓN DE SERVICIOS

Usualmente utilizado por el atacante con la intensión de extorsionar a las compañías para obligarlas a cumplir sus demandas y así detener el ataque. Las técnicas para intentar la denegación de servicios a servidores de bases de datos, suelen recurrir a la sobrecarga de la memoria y la CPU realizando el envío de consultas las cuales provocarán la caída del servidor [1], [3].

Las técnicas para mitigar este tipo de ataques son:

- Evitar trabajar con los puertos conocidos, 1434 para MS SQL Server; esto no evitará el ataque pero si generará confusión en el atacante al no encontrar el servicio ejecutándose en el puerto por omisión.
- Realizar el bloqueo de puertos innecesarios tanto en el servidor de bases de datos como en los de más servidores de la red utilizando medidas como firewalls físicos, equipos hardware y lógicos, aplicaciones firewall instaladas en los servidores [15].
- Si el proveedor de servicios de internet, ISP, cuenta con medidas que ayudan a mitigar el exceso de tráfico dirigido a la red propia, hacer uso de estas [3].

X. EXPERIENCIA Y EDUCACIÓN LIMITADA EN SEGURIDAD

El factor humano, los administradores de sistemas, los trabajadores de la compañía, todos ellos deben ser conscientes de la política de seguridad corporativa y comprender lo que podría ocurrir a la información si esta es robada por terceros mal intencionados. El factor humano, debe ser educado al respecto, la compañía debe realizar el esfuerzo para generar cultura organizacional y educar así a sus colaboradores en la seguridad de la información [1]

XI. CONCLUSIONES

La base de datos es sin lugar a dudas uno de los servidores más importantes dentro de la infraestructura corporativa, sin más es allí en donde es almacenada la información vital: información contable (cuentas por cobrar, cuentas por pagar, cartera pendiente, información tributaria), información de clientes (datos de tarjetas de crédito, histórico de compras realizadas, información de contacto), información de proveedores (compra y venta de insumos). Por lo anterior este servidor se convierte en blanco de ataques cuya finalidad es hacerse con el control del mismo para, mediante técnicas de consulta y extracción de información, lograr el acceso a la misma y usarla con fines delictivos. Con base en lo anterior, se hace necesario desarrollar una política de seguridad acorde que permita al personal encargado la correcta toma de decisiones en la configuración y administración de este servidor; de manera que al presentarse un incidente de seguridad se pueda responder de manera adecuada tanto para evitar la fuga de información como para facilitar la recolección de evidencia de los hechos sucedidos.

De acuerdo a esto, se hace necesario definir dentro de la política de seguridad estrategias como el cifrado de los medios de copia de seguridad, seguimiento a condiciones de auditoría anormales (consultas a la base de datos, conexiones en horarios no laborales, intentos de conexión de usuarios administradores), el manejo del almacenamiento de los registros de auditoría (tiempos de almacenamiento y disposición de los mismos), considerar el cifrado de las base de datos importantes a la compañía y la metodología a seguir para la creación de los usuarios. Adicionalmente se deben definir las pautas a seguir para una correcta configuración e instalación de las bases de datos antes de colocarlas en ambientes de producción, esto es cuentas de servicio, permisos de acceso a los usuarios y configuraciones de instalación. Finalmente y no menos importante se debe considerar al factor humano, este debe ser capacitado en temas de seguridad para así lograr comprender los riesgos a los que se expone la información.

Finalmente se aclara que si bien en el documento se hizo uso del motor de base de datos MS SQL Server 2012 de Microsoft y los ejemplos de código Transact-SQL estuvieron enfocados en esta, las medidas pueden ser tomadas y empleadas por un administrador de base de datos para motores de otros fabricantes.

II. AGRADECIMIENTOS

A mi familia y novia por su incondicional apoyo en la realización de mis labores académicas. A mis docentes de los cuales he aprendido y tomado los mejores ejemplos de disciplina y constancia. Muchas gracias a todos

REFERENCIAS

[1] Imperva (2013). Top ten database threats [En Linea]. Disponible: http://goo.gl/mwknzN
[2] J. Forristal. (1998). NT Web Technology Vulnerabilities. [En Linea].Disponible:http://phrack.org/issues/54/8.htm
[3] Verizon. (2014). Data Breach Investigations Report. [En Linea]. Disponible: http://goo.gl/TUepPc
[4] S. Rohilla. (2013, Nov). Database Security by preventing SQL injection Attacks in Stored Procedures. [En Linea]. Disponible: http://goo.gl/uLcLBa
[5] S. B. Suddeth. (2002, Ene.). Database thefinal firewall. *Information Security Reading Room*. [En Linea]. Disponible. http://goo.gl/sveqYR.
[6] B. Beauchemin. (2012, Ene). SQL Server 2012 Security Best Practices - Operational and Administrative Tasks.[En Linea]. Disponible: http://goo.gl/V18K6
[7] Microsoft. (2014, Jul).SQL Server 2008 Security Overview for DatabaseAdministrators.[En Línea].Disponible: http://goo.gl/7Q8n0q
[8] J. Joshi. (2013, Ago).An adaptive risk management and access control framework to mitigate insider threats. [En Línea]. Disponible: http://goo.gl/QVGgyk
[9] Microsoft. SQLCMD Utility. [En Línea]. Disponible: http://goo.gl/atHm3b
[10] Microsoft. Conectarse a datos de SQL Server (importar). [En Línea]. Disponible: http://goo.gl/8XDKOV
[11] Microsoft. Referencia de Transact-SQL (Transact-SQL). [En Línea]. Disponible: http://goo.gl/sc77mN
[12] Microsoft. StoredProcedures (DatabaseEngine).[En Linea]. Disponible: http://goo.gl/flk5ns
[13] R. Langner. (2013, Nov). ToKillACentrifugue. [En Línea]. Disponible: http://goo.gl/uX3mJG
[14] Microsoft. Help prevent malware infections on your PC. [En Línea]. Disponible: http://goo.gl/7ZKUJ4
[15] D. Winner. Making your network safer for databases.[En Línea]. Disponible: http://goo.gl/SOZvC7
[16] D. Kiely. SQL Server 2012 Keeps your data a Little more secure. [En Línea]. Disponible: http://goo.gl/nfglPL
[17] Microsoft. (2014). Transparent Data Encryption. [En Línea]. Disponible: http://goo.gl/aH2lQI
[18] B. Guhanik. (2002, Ago.). Service accounts vulnerabilities. Information Security Reading Room.[En Línea].Disponible: http://goo.gl/dIUIRY
[19] M. Russinovich, Windows Internals sixth edition part 1. Washington: Microsoft Press, 2012, pp. 310-312.

Miguel José Garcia Carvajal received the Engineering degree in Systems Engineering from Universidad Cooperativa de Colombia, Bucaramanga, Colombia, in 2010. He has been working as a software Enginner since 2010, first at Vanguardia Liberal and then at Marval, where is currently working since october 2013. Recently he had finished his studies as a Security Specialist and he will get his degree on the next September.