PROPUESTA Y VALIDACIÓN DE METODOLOGÍA PARA ASEGURAR LA INFORMACIÓN EN UNA BASE DE DATOS

DANIEL RICARDO GONZALES ESTRADA

LISDEY YUBEINY ZAPATA JIMENEZ

YENNIFER CRISTIN VILLA BECERRA

INSTITUCION UNIVERSITARIA ITM

FACULTAD DE INGENIERIAS

INGENIERIA DE SISTEMAS

MEDELLIN I - 2018

PROPUESTA Y VALIDACIÓN DE METODOLOGÍA PARA ASEGURAR LA INFORMACIÓN EN UNA BASE DE DATOS

DANIEL RICARDO GONZÁLES ESTRADA

LISDEY YUBEINY ZAPATA JIMÉNEZ

YENNIFER CRISTIN VILLA BECERRA

Trabajo de grado para obtener el título de Ingeniero de Sistemas

Asesor: GUSTAVO MACÍAS SUÁREZ

INSTITUCION UNIVERSITARIA ITM

FACULTAD DE INGENIERIAS

INGENIERIA DE SISTEMAS

MEDELLIN I - 2018

Nota de aceptación

_____

_____

_____

_____

_____

_____

_____

Presidente del Jurado

_____

Jurado

_____

Jurado

_____

Medellín, Mayo 3 de 2018

Resumen

Este trabajo de grado propone una metodología de seguridad en base de datos, la cual está enfocada en una serie de procedimientos, técnicas y métodos soportados y documentados en este escrito, cuya finalidad es exponer una reducción en la brecha de vulnerabilidad en la seguridad de la información de base de datos.

Como punto de partida se llevó a cabo una investigación de los métodos de seguridad actuales y más eficientes documentados que se aplican en los motores de base de datos SQL Server, MySQL y Cassandra, la investigación tomo como premisa hallar la manera más adecuada posible de asegurar los datos almacenados en una BD y restringir o denegar el acceso a agentes internos o externos, con base en este planteamiento se implementaron los métodos de control de acceso y cifrado de datos, los cuales constan de una serie de procesos que fueron ejecutados y probados en los tres motores de bases de datos mencionados anteriormente para asegurar la información.

Con el fin de comprobar la efectividad de la seguridad implementada, se simularon diversos ataques al control de acceso y al cifrado de datos, simulando redes de área domestica (LAN), y conexiones directas en un mismo nodo, por medio de diferentes aplicaciones y software especializado en aprovechar este tipo de vulnerabilidades.

En la pruebas y ataques realizados a las bases de datos se logró identificar algunos resultados sobre las aplicaciones más efectivas para vulnerar la seguridad de las bases de datos, entre ellos se encuentran en el sistema operativo “Kali Linux”, además a esto se pudo identificar el motor de base de datos que proporciona mayor integridad y confidencialidad a la información almacenada es SQL Server, ya que en este se puede implementar un mayor número de métricas que permiten elevar el nivel de seguridad de los datos almacenados.

Reconocimientos

La realización de este proyecto de grado fue posible, en primer lugar, a la cooperación brindada por el Profesor Jorge Iván Bedoya Restrepo y Gustavo Hernán Macías, a quienes agradecemos su ayuda constante y acompañamiento durante el proceso; además al comité de trabajos de grado de la facultad de ingeniería de sistemas por permitirnos la oportunidad de desarrollar la propuesta e investigación planteada, por su dirección y enfoque en el concepto del resultado final.

Como es de entender, se agradece a todos los autores, investigaciones, artículos y pruebas de las cuales nos nutrimos de información para la investigación y experimentación.

Agradecemos a nuestro grupo de trabajo, familia, amigos y colegas que de forma directa e indirecta fomentaron un continuo estimulo durante todo el proceso hasta al final de este.

Acrónimos

SQL Structured Query Language.

CQL Cassandra Query Language.

db Base de datos.

priv Privilegios.

RBAC Control de Acceso Basado en Roles.

SMK Clave Maestra de Servicio.

DMK Clave Maestra de base de datos.

3DES triple des, algoritmo de cifrado.

AES Advanced Encryption Standard, algoritmo de cifrado.

md5 Message-Digest Algorithm 5, algoritmo de cifrado.

SHA o SHA1 Secure Hash Algorithm, algoritmo de cifrado.

SSL Secure Socket Layer, protocolo criptográfico.

KS KeyStore, almacén de certificados con claves privadas.

keepass KeePass Password Safe, software para almacenar contraseña.

cqlsh es el Shell de línea de comandos de Cassandra, con el cual se pueden ejecutar instrucciones en el lenguaje de consultas CQL.

column family contenedor para filas.

Keyspace contenedor de espacio de nombres (db)

sa system administrator (usuario administrador por defecto de base de datos SQL Server)

SP1 Service Pack 1, paquete de actualización.

SP2 Service Pack 2 paquete de actualización.

TB Terabyte.

MyISAM Indexed Sequential Access Method, Método de Acceso Secuencial Indexado.

NoSQL bases de datos no relacionales.

role Conjunto de permisos asignados a usuarios.

localhost contexto de redes.

PING Packet Internet Groper, buscador de paquetes en redes.

Script archivo de procesamiento por lotes.

SGBD sistema gestor de base de datos.

DBA Administrador de base de datos.

hash Algoritmo matemático para transformar datos con una longitud fija.

Root cuenta superusuario (permite el acceso absoluto a todo el sistema).

Kali Linux es un sistema operativo basado en Debian GNU/Linux.

HexorBase aplicación de base de datos, diseñada para administrar y auditar múltiples servidores de bases de datos simultáneamente.

Crunch Es un generador de listas de palabras.

Cain & Abel es una herramienta de recuperación de contraseña.

RBAC (Role Based Access Control) Control de Acceso Basado en Roles.

Salt texto generado aleatoriamente para generar un hash más seguro.

TABLA DE CONTENIDO

Índice de ilustraciones

Índice de Tablas

1. Introducción

Desde el primer momento en que se inició a almacenar información importante en medios virtuales, la protección de información se ha convertido en un aspecto muy importante en cualquier organización, es por ello que nace la necesidad de mejorar los esquemas, métodos y procesos de seguridad establecidos, especialmente los que se refieren a la información almacenadas en una base de datos, esto conlleva a generar una cultura de seguridad de la información basada en la importancia de proteger los datos con una serie de recomendaciones específicas para cada segmento o área intervenida, aplicando políticas estrictas que permitan mitigar las vulnerabilidades a las que puede estar expuesta una base de datos.

Considerando la importancia que se tiene actualmente de reducir el nivel de exposición de la información en las organizaciones, el presente proyecto propone una metodología que permita aumentar el nivel de seguridad de la información almacenada en una base de datos, a través del análisis y la experimentación de los métodos y buenas prácticas de seguridad aplicables a los motores de base de datos tales como: SQL Server, MySQL y Casandra, para de esta forma lograr crear una propuesta estructurada del proceso que pondrá en funcionamiento la metodología en un sistema gestor de base de datos.

Objetivos

Objetivo General.

Implementar una metodología que permita asegurar la información en una base de datos, por medio del análisis de los métodos de seguridad existentes y de las experimentaciones en tres motores de base de datos distintos, generando la estructura del proceso para la implementación y operación de la metodología en un sistema gestor de base de datos.

Específicos

- Investigar cuáles son los métodos de seguridad que existen actualmente en el mercado.
- Determinar cuáles son los ataques a las bases de datos que ya se encuentran identificados actualmente en el mercado.
- Realizar experimentaciones implementando los métodos de seguridad encontrados en tres motores de base de datos: SQL Server, MySQL y Cassandra, con el fin de realizar ataques con la finalidad de infringir la seguridad de cada uno.
- Realizar el análisis de los métodos de seguridad teniendo en cuenta los resultados de la experimentación y de los ataques.

2. Marco Teórico

Se conoce a una base de datos como una colección de datos relacionados que se almacenan en un sistema de información para ser utilizados por diferentes usuarios y aplicaciones. Los componentes más importantes a utilizar son los datos, el sistema gestor de base de datos (SGBD) y los usuarios, el conjunto de estos componentes es la base de datos; la integración de los datos conforma la información a utilizar por el usuario y el SGBD es la interfaz entre el usuario, las aplicaciones y la BD.

A través del SGBD el usuario puede interactuar con la información almacenada, consultando, modificando, eliminando o ingresando nuevos datos.

Existen diferentes tipos de usuarios: los que utilizan las aplicaciones que se conectan a la BD, los que consultan directamente en el SGBD utilizando un lenguaje de consulta estructurada, y los que se encargan de administrar y gestionar todas las actividades relacionadas con la BD, a este último se le conoce como administrador de base de datos (DBA). (Yera, 2007)

Para lograr reducir la posibilidad de fallas de seguridad, perdida de información o incidentes que pongan en riesgo la información almacenada, en la actualidad se cuenta con métodos de seguridad que varían de acuerdo con el motor de base de datos que se esté usando, los métodos de seguridad ayudan a establecer ciertos servicios de seguridad como son la confidencialidad, autenticación, integridad, protección a la réplica, auditabilidad o trazabilidad y accesos no autorizados, entre otros.

Los avances tecnológicos hacen que cada vez se tenga más acceso a la información de forma remota, y permiten realizar gran cantidad de procesos electrónicos que llevan información

importante de cada usuario; tal como son manejo de redes sociales, transacciones bancarias, transacciones comerciales, encuestas, plataformas médicas, entre otros; estas transacciones e intercambio de información a través de dispositivos electrónicos convierten a los usuarios y organizaciones en posibles víctimas.

Las empresas y organizaciones deben contemplar toda una serie de lineamientos, políticas y procedimientos para tener la información los más asegurada posible, y para ellos se debe tener en cuenta que no es posible llegar a un 100% de seguridad en ninguna organización, por lo cual es fundamental establecer métodos de seguridad que logren minimizar el riesgo, para esto la investigación se enfocó en los métodos de control de acceso y cifrado de datos, para evidenciar la reducción del riesgo de y el aumento del nivel de seguridad de cada base de datos, por ende se utilizan pruebas con los procesos más vulnerables, las peticiones de acceso al servidos y la transacción de datos.

Los aspectos más importantes que se deben tener en cuenta para que un sistema sea seguro son:

- Confidencialidad: acceso a la información solo a usuarios que poseen los permisos suficientes y la autorización para acceder de manera contralada a la información.
- Integridad: el conjunto de datos que conforma la información el cual no se permite modifica o eliminar sin autorización.
- Disponibilidad: es la garantía que se debe brindar para que la información se pueda consultar en cualquier momento que sea necesaria.

De igual manera también se deben identificar y conocer los procedimientos negativos que podrían causar daños a la información.

- Amenazas: en un sistema de información se refiere a cualquier suceso, persona, máquina o acceso identificado que pueden atacar al sistema y producir daños, para estos casos se deben considerar las amenazas físicas (daño en hardware, perdida de la alimentación eléctrica, factores externos, entre otros) como las lógicas (Errores de usuario, antivirus, firewall entre otras).
- Riesgos: es un suceso conocido e identificado donde se cuantifica la probabilidad de ocurrencia de la materialización de este.
- Vulnerabilidades: se consideran como la debilidad en el nivel de seguridad identificada de alguno de los componentes que puede ser utilizada para causar daño y puede venir de dispositivos físicos o transacciones lógicas en un sistema.

Los ataques suceden cuando el intruso identifica una vulnerabilidad, la aprovecha y se materializa una amenaza; las amenazas más frecuentes se encuentran en el mal manejo de la seguridad o el exceso de confianza que dan paso a abusos de privilegios.

Existen medidas para mitigar las amenazas y proteger las bases de datos, entre ellas se encuentran los principios básicos de seguridad en base de datos (Chávez, 2015); los cuales generan pautas a la hora de establecer o utilizar métodos para mejorar la seguridad en la custodia de la información; Se resaltan los siguientes:

- Identificar la sensibilidad: es necesario tener identificados los datos sensibles de las bases de datos y automatizar el proceso de identificación para estar preparado en caso de cualquier cambio en los sistemas de información y asegurarlos contra software malicioso o malware.

- Evaluación de la vulnerabilidad y la configuración: se evalúa la configuración inicial de la instalación de la base de datos y del sistema operativo, los archivos con parámetros de configuración, programas ejecutables y la versión del SGBD para validar que no contenga vulnerabilidades conocidas como, una instalación estándar o permitir realizar consultas SQL desde las aplicaciones o capa de usuarios; ante estas situación el DBA puede limitar y delimitar el acceso a los procedimientos y datos para ciertos usuarios, respectivamente.

- Endurecimiento: en este proceso se compromete la eliminación de todas las funciones y opciones que no se utilizan, establecer y aplicar una política estricta de todo lo que se puede y no se puede hacer, además de los resultados de la evaluación de la vulnerabilidad mencionada anteriormente.

- Auditar: es el proceso de realizar autoevaluaciones frecuentes a sus procesos y políticas aplicadas en la realidad, con el fin de tener un seguimiento a las recomendaciones para asegurar que no se desvié del objetivo principal, la seguridad. También se debe automatizar el control de la configuración para identificar cualquier cambio, utilizando sistemas de alertas sobre modificaciones en esta.

- Monitoreo: realizar una supervisión en tiempo real y dinámica de la actividad en la base de datos que permita limitar su exposición e identificar procesos inusuales que puedan provocar ataques, cambios no autorizados en los datos, o en los privilegios de las cuentas y en la configuración. Además, la supervisión de los usuarios con permisos privilegiados ayuda a detectar intrusiones ya que los ataques más comunes se hacen con usuarios de este nivel.

Para llevar a cabo la supervisión se pueden adquirir agentes inteligentes de monitoreo y detección de intrusiones para uso indebido.

- Pistas de Auditoría: es el proceso de realizar una trazabilidad a las actividades que ejecutadas dentro del SGBD y que hagan uso de la información, en este aspecto también se pueden implementar agentes inteligentes los cuales ejecutan las tareas programadas y utilizan como insumo los logs, generados por la trazabilidad.

- Autenticación, control de acceso y gestión de derechos: hace referencia al control de acceso de los usuarios, la autenticación y administración de los privilegios asignados a cada usuario, es importante implementar reportes de los permisos de usuarios y revisarlos periódicamente.

  Cifrar los datos sensibles evita que los usuarios accidentalmente puedan consultar los datos e interpretar la información.

2.1 Métodos de Seguridad

Además de los principios básicos de seguridad en bases de datos, también es importante conocer los métodos de seguridad que ayudan a mitigar el riesgo a los ataques más comunes, que son reconocidos actualmente y que vulneran la información contenida en una base de datos; por consiguiente, se implementan en el medio los siguientes métodos:

Control de acceso: se define como el proceso de autorizar a los usuarios, grupos y equipos el acceso a la red o el sistema (Microsoft, 2018)

Se basa principalmente en la creación de cuentas de usuario y contraseñas que son administradas y controladas a través del SGBD, el DBA es quien a su vez administra el sistema y tiene una cuenta privilegiada en el SGBD con permisos no disponibles para otras cuentas ordinarias, entre ellas creación de cuentas de usuario, asignación, revocación de privilegios y niveles de seguridad.

Uno de los controles que los DBA pasan por alto es el de permitir el acceso Root o cuenta superusuario (permite el acceso absoluto a todo el sistema) con los valores que vienen por defecto en el SGBD, esto crea fallas o huecos en la seguridad, de igual manera es necesario eliminar bases de datos utilizadas como pruebas o basura y la cuenta de prueba creada durante la instalación inicial del SGBD, también se recomienda revisar periódicamente los permisos otorgados a los usuarios, la cantidad de usuarios y asegurar que no tengan modificaciones no autorizadas, la debilidad en este nivel (humano) permite burlar las otras medidas de seguridad establecidas.

Algunos métodos de control importante para tener en cuenta son:

- Control de acceso a usuarios: hace referencia a los permisos que se le van a asignar a cada usuario del sistema.
- Control de Inferencia: Se basa en bases de datos estadísticas que solo permite que se pueda acceder a información estadística de un usuario.
- Control de flujo: Controla que el flujo de información no afecte los objetos de menor protección dentro de la base de datos.

- Revocación de privilegios: Se utiliza cuando se asigna temporalmente a un usuario privilegios para realizar una actividad determinada dentro de la base de datos, después de realizar dicha actividad el privilegio debe de ser revocado.
- Control de accesos utilizando Triggers: Controla que se ejecute un comando no válido por un usuario, ya que el trigger se ejecuta para proteger e impedir que se realice algún cambio a una tabla. (Avilés, 2015)

Cifrado de datos: proceso que consiste en convertir la información de la base de datos en datos incomprensibles mediante la utilización de un algoritmo de cifrado, permitiendo proteger los datos de una interceptación en el transporte de paquetes por medio de la red o la intrusión de un usuario no autorizado para que así se pueda mantener la integridad y la confidencialidad de estos.

Existen dos sistemas de cifrado, Simétrico y Asimétrico

- Cifrado Simétrico: en este proceso se utiliza una misma clave para cifrar y descifrar el mensaje, las dos partes acuerdan cuál será la clave de encriptación, el emisor envía el mensaje por un canal que puede ser seguro o no seguro, cifra el mensaje con el algoritmo de encriptación y la clave, el receptor descifra el mensaje con un algoritmo de desencriptación y la misma clave para visualizar el mensaje.

  La debilidad de este sistema reside en que las dos partes comparten la misma clave y que el mensaje puede viajar por un canal no seguro, un atacante podría interceptar y descifrar los mensajes, por ello es importante que el algoritmo de encriptación se diseñe con un

amplio rango de claves para que un atacante no pueda probar todas las posibilidades, logrando dar con la clave y así desencriptar el mensaje.

- Cifrado Asimétrico: a diferencia del cifrado simétrico, este utiliza dos claves para cifrar y descifrar, las dos claves forman un par único, no es posible que dos personas obtengan el mismo par de claves.

  El emisor envía el mensaje a través de un canal no seguro y cifra el mensaje con la clave pública del receptor; el receptor descifra los datos con el algoritmo y la clave privada que solo esté conoce.

  Este sistema es más seguro debido a que una vez cifrado el mensaje con la clave pública sólo el destinatario puede descifrar el mensaje, a pesar de que el canal no sea seguro. (Gutierrez del Moral, 2013)

2.2 Ataques informáticos

Para lograr un ataque informático o en un caso específico, un ataque a los métodos de control de acceso y cifrado de datos, los intrusos deben de disponer de medios técnicos, conocimientos y herramientas que permitan vulnerar la seguridad del sistema.

Existen diferentes tipos de ataques informáticos, los activos que intentan alterar los recursos del sistema o afectar su operación y los pasivos, que violan la confidencialidad sin afectar el estado del sistema, con el fin de hacer uso de la información almacenada, además existen otro tipo de ataques denominados ataques internos, los cuales son producidos por un agente interno, el cual está autorizado para acceder a los recursos del sistema y los usa de forma inadecuada y los ataques

externos, son producidos por un agente externo al grupo de usuarios permitidos y con privilegios en el sistema que realiza una intrusión para extraer información, dañarla o en el peor de los casos cifrarla de tal manera que se vuelva inservible. (Moral L. G., 2013)

Los ataques pasivos utilizan las siguientes técnicas:

- Ataque de diccionario de datos: utiliza un archivo de texto denominado diccionario que contiene un listado de posibles contraseñas y/o usuarios, consiste básicamente en un método de prueba y error que tiene como propósito validar con cuál de las claves registradas en el diccionario se logra acceder al sistema.

  En la web se pueden encontrar diferentes tipos de diccionarios, con una gran cantidad de registros, combinaciones de números, letras, caracteres especiales, con diferente longitud de cadena y con claves comunes, las que por facilidad suelen usar los usuarios como son: 123, abcd,4321, entre otros.

- Ataques de fuerza bruta y cracking: al igual que el anterior utiliza el método de prueba y error para intentar descifrar clave y adicionalmente, un algoritmo de descifrado para intentar descifrar una clave que se encuentre encriptada. (Moral L. G., 2013)

- Inyección de código SQL: es un ataque donde se inserta código malicioso en las cadenas que se pasan a la instancia de SQL del motor de base de datos para el análisis y ejecución. Una inyección de código SQL se basa en insertar directamente código en las variables que el usuario ha especificado en un formulario de una aplicación web. (Microsoft, 2016)

Para ejecutar un ataque informático se debe disponer de las herramientas adecuadas, las cuales existe una gran variedad en la web, algunas son gratuitas, otras de software libre y otras de pago por licencia, para nuestro proyecto tuvimos en cuenta algunas para poder identificar sus ventas y desventajas frente a nuestro objeto de investigación: Algunos de los programas utilizados fueron:

Kali Linux: es un sistema operativo basado en Debian GNU/Linux diseñado principalmente para la auditoría y seguridad informática en general, cuenta con diferentes aplicaciones preinstaladas para poner a prueba la seguridad de los sistemas, entre ellas se incluyen las siguientes:

- Crunch: Es un generador de listas de palabras donde se puede especificar la longitud y el conjunto de caracteres para que se cree un archivo con todas las combinaciones y permutaciones posibles.

  Para crear un diccionario se ingresa la cantidad mínima de caracteres, la cantidad máxima, las opciones que pueden ser letras, números y caracteres especiales y el nombre del archivo.

  Ejemplo de sintaxis:

  Crunch <min> <max> [opciones] > nombre_archivo.txt

  (bofh28, 2014)
- HexorBase: es una aplicación de base de datos diseñada para administrar y auditar múltiples servidores de bases de datos simultáneamente desde una ubicación centralizada, es capaz de realizar consultas SQL y ataques de fuerza bruta contra servidores de bases de datos comunes (MySQL, SQLite, Microsoft SQL Server, Oracle, PostgreSQL). (Ekiko, 2014)

Con esta aplicación se realiza un ataque de diccionario, especificando la dirección IP del servidor, el puerto y los diccionarios de Usuarios y Claves.

- Johnny: Es una aplicación con interfaz gráfica que desencripta hash de contraseñas. (Shinnok, 2014)

  Para realizar un ataque con esta aplicación se crea un archivo de texto con los hashes de contraseñas que se deseen desencriptar, se carga el archivo y se inicia el ataque, al finalizar se podrá observar cual es el valor real de cada contraseña.

Cain & Abel: es una herramienta de recuperación de contraseña para los sistemas operativos de Microsoft. Permite recuperar fácilmente varios tipos de contraseñas, descifrando contraseñas cifradas usando técnicas de diccionario, fuerza bruta y criptoanálisis, su objetivo principal es la recuperación simplificada de contraseñas y credenciales de diversas fuentes. (Montoro, 2014)

Para llevar a cabo un ataque, además de las técnicas y herramientas mencionadas anteriormente se debe realizar una prueba de intrusión o test de intrusión, la cual consiste en una prueba de vulnerabilidad que tiene como objetivo evaluar el estado de los sistemas frente ataques de tipo intrusivo, donde éticamente se buscan los niveles más bajos de seguridad del sistema en cuestión. (Isec auditors, 2017), existen tres tipos:

- Test de caja negra: consiste en auditar o evaluar el sistema sin poseer ninguna información y buscar todas las posibles vulnerabilidades que se puedan explotar como si fuera un atacante real.
- Test de caja gris: se cuenta con una cantidad limitada de información, lo que facilita la realización de otras pruebas sin tener que averiguar toda la información, y permite llegar más lejos en la evaluación.
- Test de caja blanca: es el test más completo, cuenta con toda la información necesaria del sistema para evaluar todos los posibles riesgos y vulnerabilidades, probando cada uno de los servicios y configuraciones. (ACISSI, 2015)

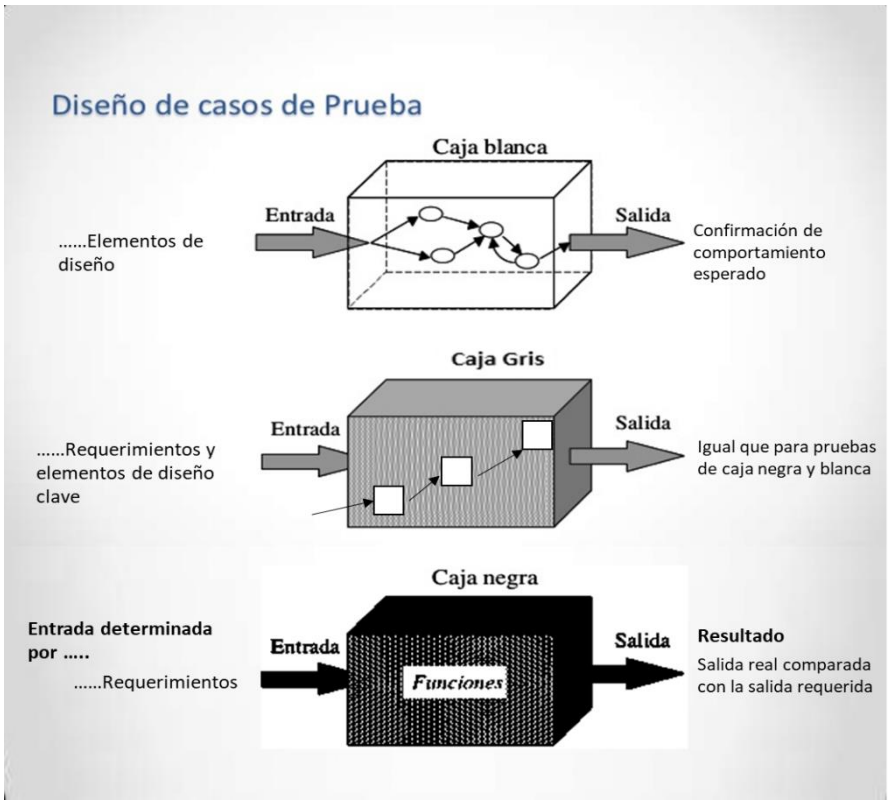

Ilustración 1: Diseño de casos de prueba

3. Metodología

Para la ejecución del proyecto se aplicó una metodología que consta de las siguientes fases:

3.1 Fase 1: Investigación de los Métodos de Seguridad en Base de Datos.

En esta fase se realizó la investigación de los métodos de seguridad en base de datos más eficientes que existen desde el año 2004 hasta la fecha, utilizando palabras claves, y un algoritmo de búsqueda en motores de búsqueda en la web y bases de datos de artículos científicos como "Science Direct", "Scopus", "Web Of Science" entre otros.

Los resultados de la investigación fueron tomados para aumentar el conocimiento en el objetivo del trabajo de grado, con el fin de buscar prácticas, métodos y esquemas de seguridad aplicados en bases de datos comerciales y no comerciales.

Algoritmo de búsqueda:

En los motores seleccionados se utilizó el siguiente algoritmo de búsqueda con el fin de hallar documentos que en su título contuvieran las palabras:

Database AND Security AND Vulnerability

Base de datos utilizada:

Sistema de Bibliotecas ITM:

https://es.slideshare.net/jabenitez88/8realizacion-de-pruebas-14981770

A continuación se describen los métodos de seguridad en cada uno de los motores de base de datos seleccionados:

Control de acceso SQL Server: se utiliza el control de acceso basado en roles, que permite asignar permisos a un rol o grupo de usuarios, en lugar de a usuarios individuales.

Los roles se definen a nivel de servidor, de base de datos o de aplicación y pueden ser fijos o definidos por el usuario. (douglaslMS, 2017).

Este proceso se realiza con la creación de roles de base de datos definidos por el usuario, para ello es necesario identificar los roles del sistema, los usuarios y permisos que se van a asignar a cada rol.

Cifrado de datos SQL Server: permite realizar cifrado simétrico y asimétrico mediante la combinación de diferentes mecanismos, para ello utiliza una infraestructura jerárquica que permite cifrar por capas, donde cada capa se protege cifrando la capa anterior.

Para llevar a cabo el proceso es necesario definir los datos que se van a cifrar, el tipo de cifrado y el mecanismo a utilizar; SQL Server permite combinar diferentes mecanismos de cifrado y elegir entre varios algoritmos.

Los mecanismos de cifrado de SQL Server son los siguientes:

- Clave Simétrica: es una clave que se usa para el cifrado y descifrado, es decir, se utiliza la misma clave para cifrar y descifrar los datos
- Clave Asimétrica: es una clave que consta de una clave pública y una privada, la primera se utiliza para cifrar y la segunda para descifrar.
- Certificados: es una instrucción firmada digitalmente que enlaza el valor de una clave pública con la identidad de la persona, dispositivo o servicio que tiene la clave privada correspondiente. Se utiliza para cifrar datos o conexiones, firmar paquetes y otros objetos.

- Transact-SQL funciones: se utiliza para cifrar datos individuales a medida que se insertan o actualizan en la base de datos.
- Cifrado de datos transparente: es un cifrado especial que utiliza una clave simétrica para cifrar una base de datos completa. (Macauley, 2017)

SQL Server utiliza claves de cifrado para proteger los datos, las credenciales y la información de conexión que se almacena en una base de datos, existen dos claves principales:

- Clave Maestra de Servicio (SMK): se genera automáticamente la primera vez que se inicia la instancia de SQL Server, se utiliza para cifrar una contraseña de servidor, las credenciales y la clave maestra de base de datos. Aplica para una instancia de SQL Server.
- Clave Maestra de base de datos (DMK): es una clave simétrica que se utiliza para proteger las claves privadas de certificados y las claves asimétricas de la base de datos. Esta clave se cifra mediante el algoritmo Triple DES y una clave proporcionada por el usuario. Aplica para una base de datos. (Macauley, 2017)

De los algoritmos de cifrado que admite SQL Server se destacan por ser los más utilizados y seguros los siguientes:

- TRIPLE DES (3DES): es un algoritmo de cifrado que usa tres claves de 64 bits cada una y tres ejecuciones del algoritmo DES. La función sigue la secuencia cifrar-descifrar-cifrar. Tiene una longitud de clave de 168 bits. (Stallings, 2004)
- AES: es un algoritmo de cifrado simétrico que utiliza una clave de 128 bits de longitud para cifrar y descifrar. La longitud de clave puede ser de 128, 192 o 256 bits. Los datos por encriptar se dividen en segmentos de 16 bytes y cada segmento se puede ver como un bloque o matriz

de 4x4 bytes al que se le llama Estado, este se va ordenando y a partir de ahí se le realizan una serie de operaciones. (González, 2015)

En la *Ilustración 2*. Se observa la jerarquía de cifrado de SQL Server y sus combinaciones más comunes.

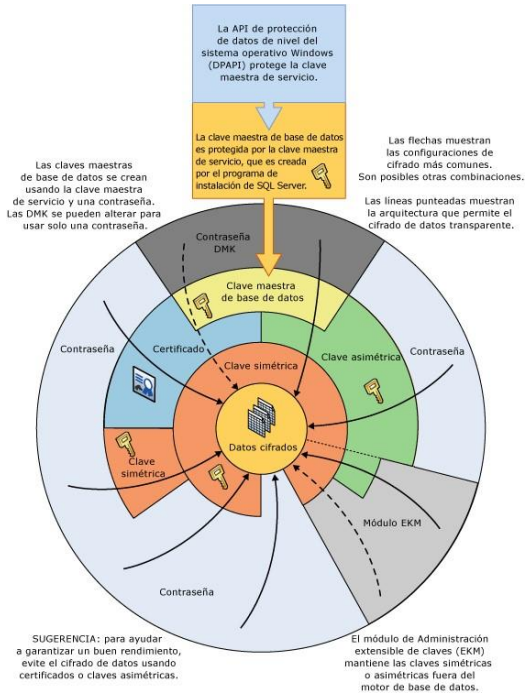

Ilustración 2. Jerarquía de cifrado.

Fuente: (Macauley, 2017)

Control de acceso MySQL: este motor cuenta con un sistema de privilegios de acceso, cuya función principal es autenticar un usuario que se conecta desde un equipo dado y asociar ese usuario con los privilegios que tiene en una base de datos.

El control de acceso de MySQL implica dos etapas: (1) el servidor comprueba si debe permitir la conexión al usuario y (2) una vez conectado, el servidor comprueba cada comando que ejecuta para ver si tiene suficientes permisos para hacerlo.

La información de los privilegios se almacena en las tablas user, db, host, tables_priv y columns_priv de la base de datos mysql.

En cada tabla se almacena la siguiente información:

Tabla User: indica si se permite o se rechaza la conexión al servidor, cualquier privilegio otorgado en esta tabla indica los privilegios globales del usuario y aplican para todas las bases de datos.

Tabla db: en esta tabla se indica un usuario a que base de datos puede acceder, desde que equipo y que privilegios se permiten a la base de datos y sus tablas.

Tabla Host: se utiliza en conjunto con la tabla db para registrar, por ejemplo, que un mismo usuario de la tabla db pueda acceder a través de diferentes equipos.

Cifrado de datos MySQL: utilizar diferentes funciones de cifrado, entre ellos se implementaron: MD5, SHA1 y AES_ENCRYPT para proteger los datos almacenados.

- MD5: la función recibe como parámetro el texto a cifrar; realiza el cifrado de los datos utilizando el algoritmo MD5, mediante la suma de comprobación de 128 bits de una cadena, el valor de retorno es una cadena de 32 dígitos hexadecimales.

- SHA1: la función recibe como parámetro el texto a cifrar; realiza el cifrado utilizando el algoritmo SHA1, mediante la suma de comprobación de 160 bits de una cadena, el valor de retorno es una cadena de 40 dígitos hexadecimales.
- AES_ ENCRYPT: la función recibe dos parámetros, el texto a cifrar y el Password. Utiliza el algoritmo AES para cifrar los datos mediante una clave privada que se requiere para encriptar y desencriptar.

Control de acceso Cassandra: se utiliza el control de acceso basado en roles (RBAC), por medio de funciones de base de datos, que pueden representar un usuario o un grupo de usuarios, aplica para la administración de autenticación y de permisos, facilita la administración de permisos al agrupar privilegios en roles que se pueden asignar a usuarios específicos. (Apache Software Foundation, 2016)

En cassandra una base de datos se denomina keyspace.

Cifrado de datos Cassandra: este motor de base de datos proporciona una comunicación segura entre una máquina cliente y un clúster de base de datos y entre nodos dentro de un clúster. La habilitación del cifrado garantiza que los datos se transfieran de forma segura.

Cassandra no provee una función para el cifrado de los datos almacenados en las tablas de un Keyspace, Cassandra internamente almacena las contraseñas de los roles cifrados utilizando un método hash con un salt, pero este no está disponible para la interacción y utilización del usuario o DBA.

De acuerdo a lo anterior y a diferencia de los motores anteriores, en cassandra no se realiza el método de cifrado de datos en la fase de experimentación, el escenario de cifrado es diferente al propuesto para el desarrollo del proyecto.

3.2 Fase 2: Experimentación

Con base en la información consultada en la fase 1 se llevó a cabo la fase de experimentación, en la cual se realizaron las siguientes actividades:

- Instalación de los motores de base de datos: SQL Server, MySQL y Cassandra sobre diversas plataformas Microsoft como “Microsoft Windows 10” y “Microsoft Windows 7” en sus versiones más estables.
- Ejecución de los métodos de seguridad en cada uno de los motores de bases de datos seleccionados: métodos control de acceso y cifrado de datos.
- Intentos de vulnerabilidad y simulación de ataques a los métodos de seguridad implementados en cada uno de los motores, por medio del uso de aplicaciones como: SQLPing v3.0, Cain & Abel,HexorBase y Johnny de Kali Linux; adicionalmente para SQL Server se desarrolló y ejecuto un script para descifrar los datos almacenados en las tablas.

En esta fase se logró identificar resultados que guiaron en la selección de las aplicaciones más efectivas, para aumentar la exposición al riesgo de vulnerabilidad de la seguridad aplicada, a continuación, se expone el proceso realizado:

3.2.1 Experimentación métodos de seguridad y ataques.

Para la implementación de la seguridad en las bases de datos se seleccionan tres motores de base de datos (SQL Server, MySQL, Cassandra), orientados en los métodos de control de acceso y cifrado de datos, ya que con estos se logra asegurar los procesos más vulnerables, como son el acceso al servidor y a los datos almacenados.

Dicho lo anterior, se detalla el proceso desarrollado en cada motor.

3.2.1.1Métodos de seguridad SQL Server

A. Caso práctico control de acceso: se realiza la instalación típica de SQL Server 2014, en cuanto a seguridad durante la instalación se tiene en cuenta que la clave del usuario "sa" (administrador del sistema) cumpla con las características de una contraseña segura.[1]

Se utiliza la base de datos "Adventureworks2014".

En la *Ilustración 1.* se puede observar en que parte de la instalación se asigna la clave segura para el usuario sa.

[1] Contraseña Segura: para que una contraseña sea segura debe cumplir con las siguientes características: no debe contener el nombre completo o partes significativas del nombre de la cuenta de usuario, debe estar compuesta por mínimo 6 caracteres de longitud, letras mayúsculas y minúsculas, dígitos de 0 a 9 y caracteres no alfabéticos. (Microsoft, 2017)

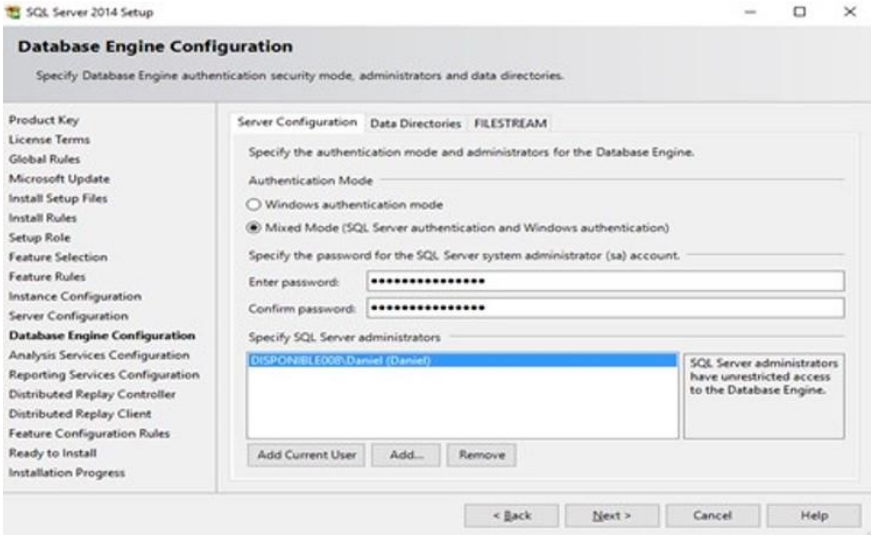

Ilustración 3.Password usuario "sa"

- Implementación control de acceso basado en roles (RBAC)

  la implementación de RBAC en SQL Server se realiza mediante la opción "Database Roles" (roles de base de datos definidos por el usuario), y es necesario establecer los roles, usuarios y permisos a utilizar.

  En la *Tabla 1*. Se encuentran los roles con sus respectivos permisos y usuarios.

  Nota: también se aplican para los motores MySQL y Cassandra

Tabla 1.Roles, permisos y usuarios

Roles	Descripción	Login	Usuario	Clave	Permisos	Tablas
Administrador	Administrador de toda la base de datos	Sa Root Cassandra	Sa Root Cassandra	rtyUI2356*.s%23	Todos	Todas

Ventas	Consulta de toda la parte de ventas	Ventas1	Ventas1	fdo6783 %.#e3	Consulta	Todas las tablas del esquema Sales
ConfigVentas	Administración de la configuración de ventas	ConfigVentas1	ConfigVentas1	eri0529,$(i9	Consulta, actualización	Todas las tablas del esquema Sales
Cumplimiento	Consulta de toda la parte de cumplimiento	Cumplimiento1	Cumplimiento1	qwcs1830 &#.v4	Consulta	Production.WorkOrder, Production.ScrapReason, Sales.ShoppingCartItem
Inventario	Administración del inventario	Inventario1	Inventario1	dfli2571) %,d6	Consulta, actualizaciòn	Todas las tablas del esquema Production
Compras	Administración de compras	Compras1	Compras1	awty1045 %$.h7	Consulta, actualizaciòn	Todas las tablas del esquema Purchasing y Person.Address

						, Person.Contact, Production.ProductVendor
Recurso_humano	Administración de personal	Recurso_humano1	Recurso_humano1	cvmn9857#%)o1	Consulta, actualizaciòn	Todas las tablas del esquema HumanResources
Presupuestos	Administración de presupuestos	Presupuestos1	Presupuestos1	iozx7239)$/b3	Consulta, actualizaciòn	Sales.Currency, Sales.SpecialOffer, Sales.SpecialOfferProduct
Produccion	Administración de producción	Produccion1	Produccion1	fwph610·&.w2	Consulta, actualizaciòn	Todas las tablas del esquema Production
Consultas	Rol de consulta para ver información	Consultas1	Consultas1	huip(#45yop.	Consulta	Sales.Store , Production.Product, Sales.Shopping

	de diferentes tablas					CartItem, Sales.SpecialOfferProduct,Sales.SpecialOffer

- Creación de roles

  Una vez identificados los roles, se procederá con la creación de estos en el motor de base de datos, en la Ilustración 2. Se visualiza como crear un nuevo rol de base de datos en SQL Server.

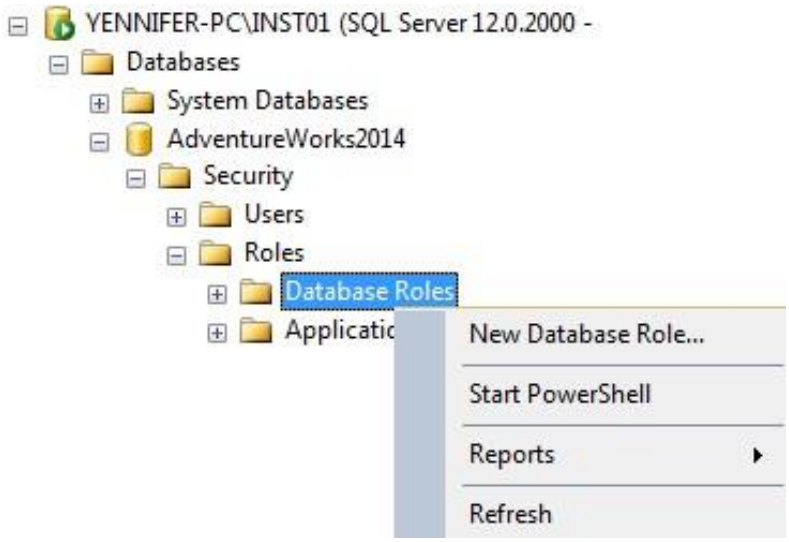

Ilustración 4. Crear nuevo rol SQL Server

Una vez se selecciones la opción “New Database Role”, aparecerá la página “General”, en la cual se ingresa el nombre del Rol, y se dejan los demás campos con los valores que tienen por defecto como se muestra en la *Ilustración 3*, finalmente se da clic en el botón “add”.

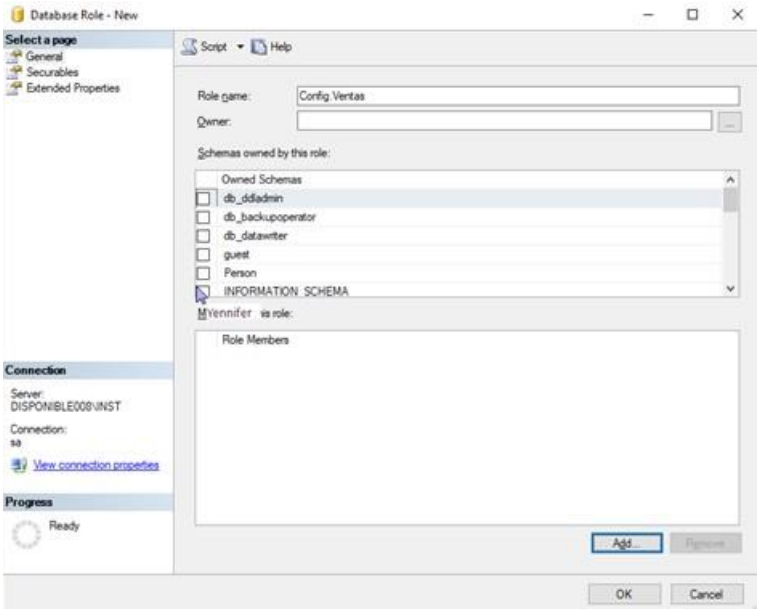

Ilustración 5. Página "General" nuevo rol SQL Server

En la página "Securable" se adicionan las tablas que puede visualizar el rol y se asigna los permisos a cada una, Ver *Ilustración 4.*

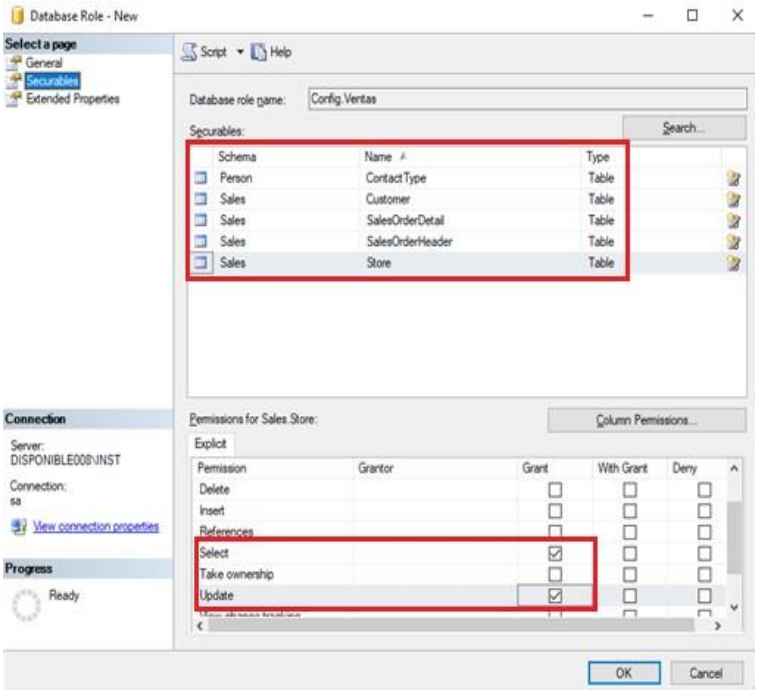

*Ilustración 6.*Página "Securable" nuevo rol SQL Server.

Para finalizar en la página “Extended Properties”, se dejan los valores por defecto y con el botón OK se crea el rol.

- Creación de login en la Ilustración 5. se crea el nombre del Login, la clave de ingreso a SQL Server, se marcan las opciones de directivas de contraseñas, se selecciona la base de datos AdventureWorks2014 y el idioma.

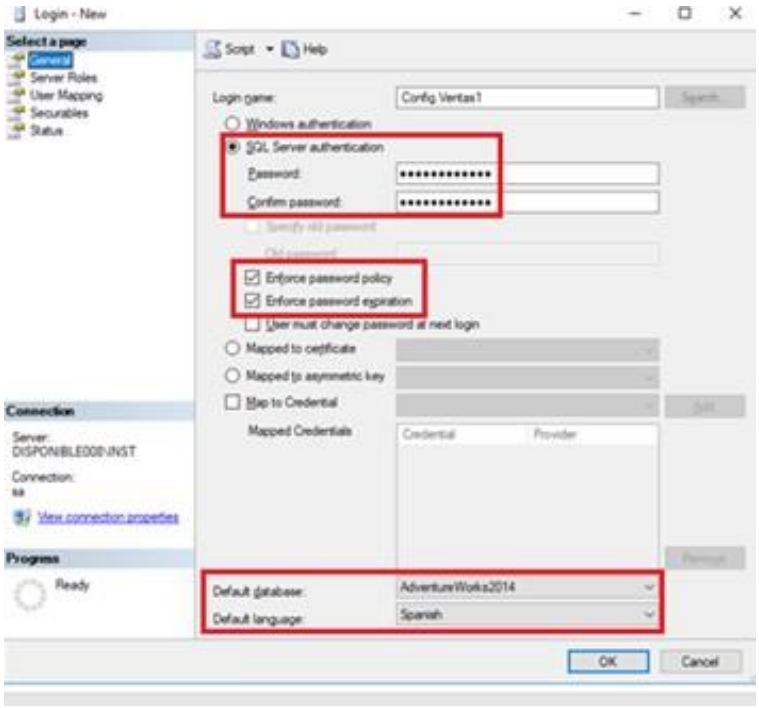

*Ilustración 7.*Página “General” nuevo login SQL Server.

En la *Ilustración 6.* En la página “Page User Mapping” se selecciona la base de datos y el rol al que va a pertenecer el usuario.

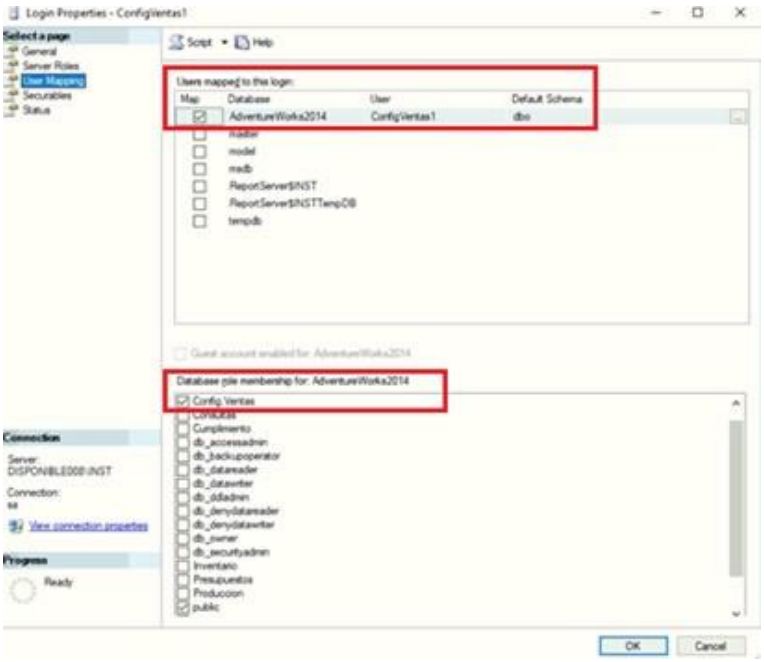

Ilustración 8. Page "User Mapping" nuevo login SQL Server.

En la *Ilustración 7*. En la página "Page Status" se marcan las opciones para otorgar permisos de conexión a la base de datos y se habilita el Login.

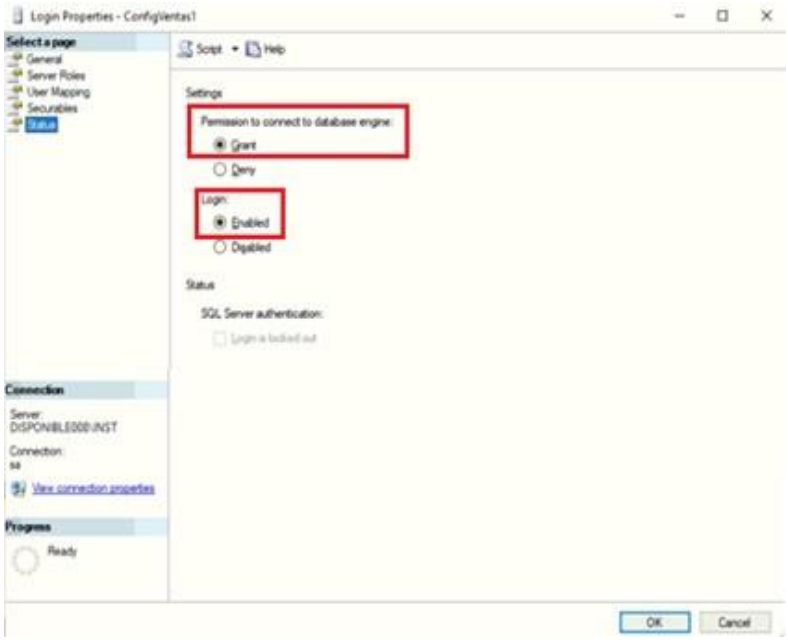

*Ilustración 9.*Página "Status" nuevo Login SQL Server.

- Asignar usuarios al rol en la Ilustración 8 se agregan los usuarios al rol que corresponden.

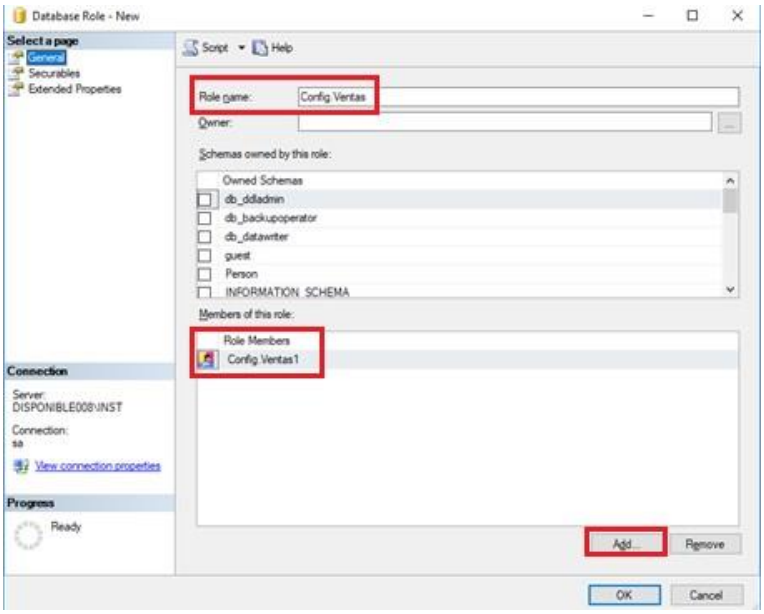

*Ilustración 10.*Asignación de usuarios al rol SQL Server

B. Caso práctico cifrado de datos: paso 1: para el proceso de cifrado de datos se analizaron cuáles de las tablas de la base de datos Adventureworks2014 contenían información delicada o susceptible para la empresa Adventure Works Cycles se eligieron tres de ellas, en cada tabla se cifro la columna que se consideró más relevante y se describen en la *Tabla 2. Tablas y columnas cifrado de datos.*

Nota: también se realiza lo mismo para los motores MySQL y Cassandra.

Tabla 2. Tablas y columnas cifrado de datos

Nombre Tabla	Nombre Columna	Justificación
HumanResources.EmployeePayHistory	Rate	Esta tabla contiene la información actual e histórica sobre los salarios de los empleados y la columna Rate indica el valor pagado por hora al empleado
Sales.CreditCard	CardNumber	Esta tabla contiene la información de las tarjetas de crédito de los clientes y en la columna CardNumber se almacena el numero completo de la tarjeta de crédito.
Production.Product	StandardCost	La tabla contiene los productos vendidos o utilizados en el proceso de fabricación de los productos vendidos y la columna StandardCost indica el costo estándar del producto

Paso 2: Para el cifrado de datos se creó un script que permitiera cifrar las columnas de las tablas mencionadas en la *Tabla 2. Tablas y columnas cifrado de datos* donde se utilizan los siguientes mecanismos:

- Clave maestra de base de datos
- Certificado
- Clave simétrica
- Algoritmo de cifrado
- Función Transact-SQL

En la *Ilustración 10.* se puede observar el script para cifrar los datos.

```
Use AdventureWorks2014
go
--Clave Maestra de Base de Datos
create master key encryption by
password ='Eip1589*B2%c$k4Q0#.rQ';

--Certificado
create certificate Cifrar1
with subject ='Cifrar';

--Creación de la clave Simétrica y cifrado con algoritmo triple_des
create symmetric key Ratekey_01
with algorithm= triple_des
encryption by certificate Cifrar1;

--Modificación de la tabla para agregar la columna que va a contener los datos
cifrados
alter table [HumanResources].[EmployeePayHistory]
add EncryptedRate varbinary (128);

--Se cifran los datos de la columna Rate y se agregan a la nueva columna
EncryptedRate
open symmetric key Ratekey_01
decryption by certificate Cifrar1;

update [HumanResources].[EmployeePayHistory]
set EncryptedRate =
ENCRYPTBYKEY(KEY_GUID('Ratekey_01'),convert(nvarchar(20),Rate))

--Se cierra la clave simétrica
close symmetric key Ratekey_01

--Se elimina la columna original Rate con los datos sin cifrar
alter table [HumanResources].[EmployeePayHistory] drop column Rate;
```

Ilustración 11.Script cifrado SQL Server

En la *Ilustración 11*. se observan las columnas con los datos cifrados después de ejecutar el script.

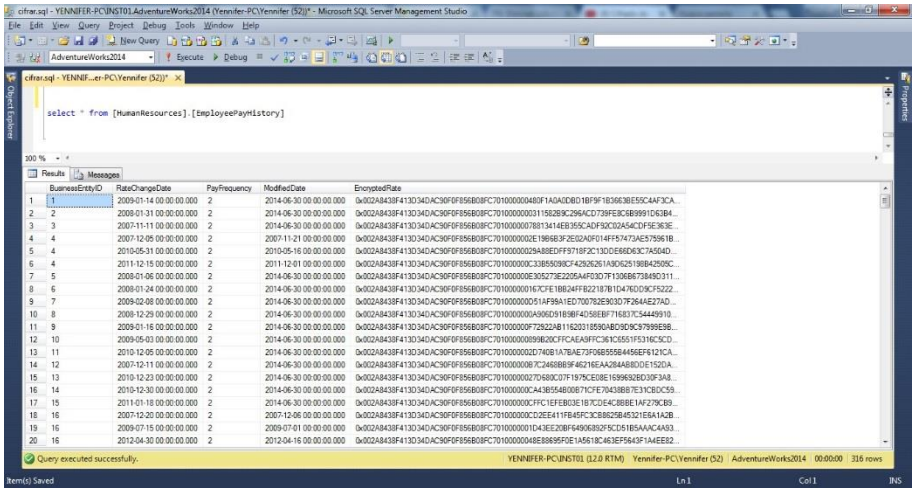

Ilustración 12.Datos cifrados SQL Server.

C. Ataques en la fase de experimentación se realizaron ataques a la base de datos AdventureWorks2014, mediante un test de intrusión de caja gris, con el fin de auditar el sistema y encontrar las falencias de seguridad, en el control de acceso (contraseñas de usuarios) y en los datos cifrados.

- Ataques control de Acceso los ataques de control de acceso se realizaron con la finalidad de obtener la clave del usuario administrador que se crea por defecto durante la instalación de cada uno de los motores de base datos, los cuales se nombran de la siguiente forma:

    - Sa: SQL Server
    - Root: MySQL
    - Cassandra: Cassandra

  Para las pruebas se asignaron claves seguras a los usuarios administradores mencionados anteriormente y se utilizó otro de los usuarios definidos en la *Tabla 1 Roles, permisos y*

usuarios y se modificó la clave asignada por una clave común, como se muestra en la Tabla 3.

*Tabla 3.*Usuarios SQL Server

Usuarios	**Claves**
Sa	rtyUI2356*.s%23
Ventas2	123

Para el proceso se utilizaron las siguientes herramientas:

- SQLPing v3.0

  Esta herramienta realiza un escaneo de SQL Server y busca contraseñas débiles mediante el ataque por fuerza bruta, listas de palabras, entre otros.

  Para realizar el escaneo con SQLPing v3.0 se suministró la dirección IP donde se encuentra instalado el servidor de SQL Server y se ejecutó el escaneo dando clic en el botón Scan.

  En La *Ilustración 12.* se observa el resultado del escaneo para la dirección Ip ingresada.

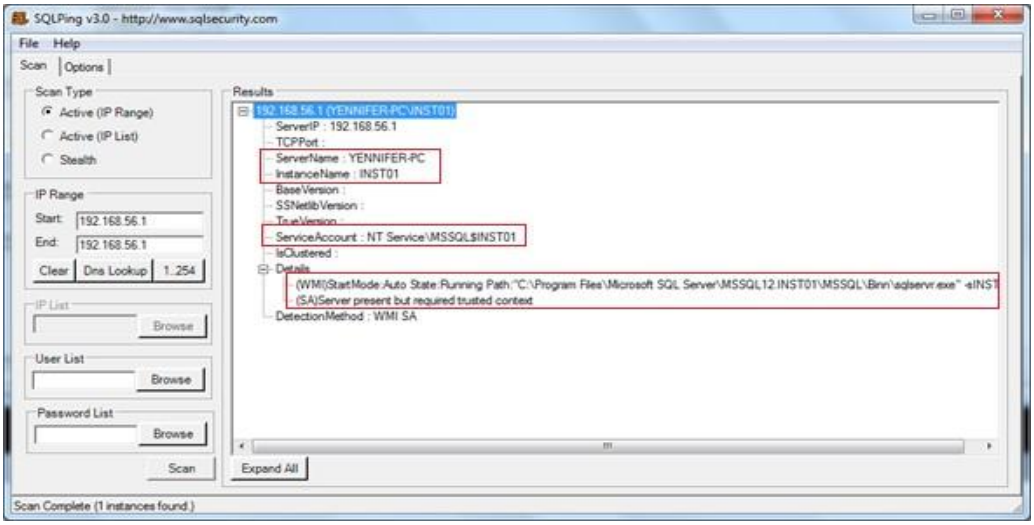

Ilustración 13. Escaneo dirección Ip SQLPing v3.0.

En el resultado se observa que la aplicación encuentra el nombre del servidor, la instancia, la cuenta de servicio y el usuario sa, sin embargo, no logra descifrar la contraseña de este, indica que el usuario se encuentra presente o activo, pero que se requiere un contexto de confianza.

Esta herramienta también permite realizar ataques de diccionario o listas de palabras; consiste en adjuntar dos diccionarios, uno de usuarios y otro de contraseñas, con el fin de hacer el escaneo y validar si con los diccionarios se consigue conexión al servidor (la herramienta realiza todas las posibles combinaciones).

En internet existen muchos diccionarios disponibles para descargar con múltiples opciones para usuarios y contraseñas, con diferente cantidad de registros, longitud, combinaciones de números, letras y caracteres especiales, etc. Generalmente contienen las claves más comunes que por facilidad suelen usar algunos usuarios (abcd,123, 4321, …).

La efectividad de un ataque diccionario radica en que la clave de acceso al servidor de base de datos se encuentre especificada en el diccionario. Entre los diccionarios más populares se encuentran: rockyou y cain & Abel.

Para realizar el ataque se utilizaron los siguientes diccionarios:

- Claves: se descargó el diccionario de claves comunes del sitio web: http://www.el-palomo.com/2012/05/se-comparte-diccionario-de-contraseas-de-2gb/
- Usuarios: se realizó un diccionario con los nombres de los usuarios Sa y Ventas2.

En la *Ilustración 13*. Se pueden visualizar los datos que contiene cada diccionario

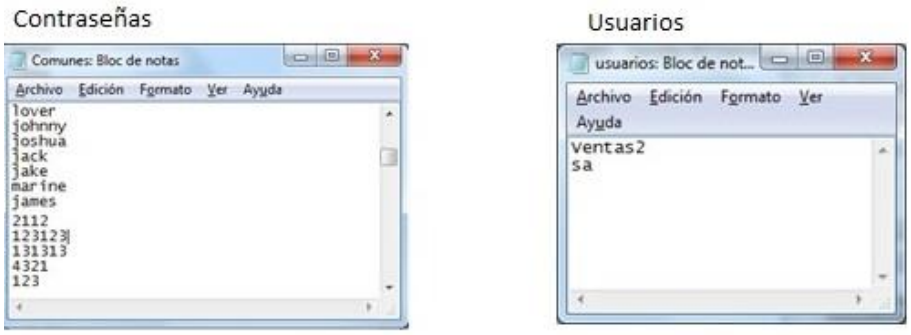

Ilustración 14. Diccionarios1 SQLPing v3.0.

En el ataque de diccionario se ingresa la IP del servidor y los diccionarios, el proceso se observa en la *Ilustración 14.*

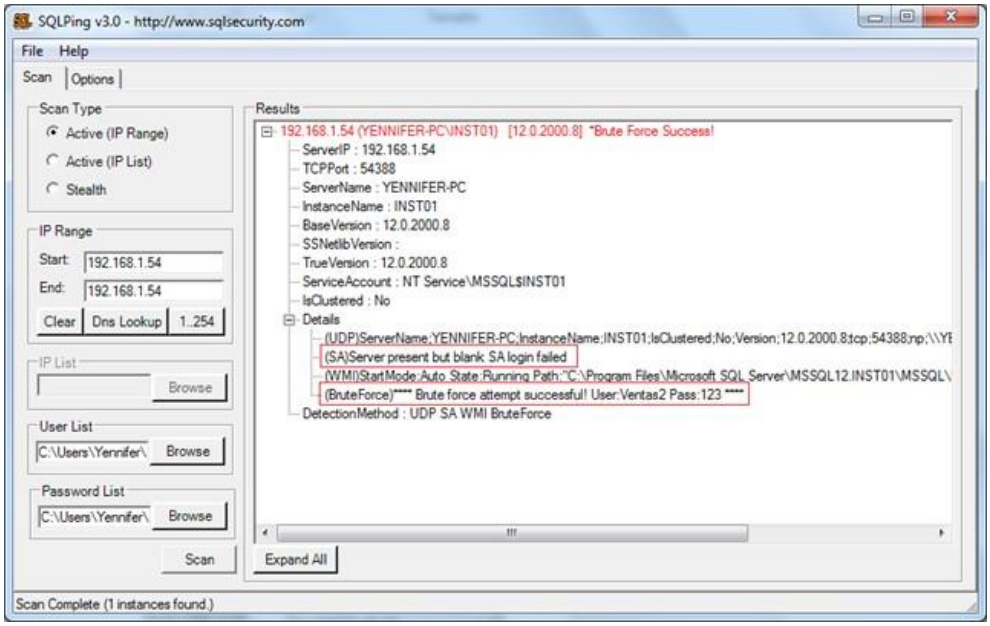

Ilustración 15.Ataque de diccionario1 SQLPing v3.0.

En el resultado se puede notar que la clave para el usuario "sa" no fue posible identificarla, en cambio para el usuario "Ventas2" si fue posible. Los resultados obtenidos se deben a que el diccionario para claves contiene el registro "123" que corresponde al usuario "Ventas2", mientras que la clave "rtyUI2356*.s%23" del usuario "sa" no existe en el diccionario.

Para que el ataque contra el usuario "sa" sea efectivo se requiere que el diccionario contenga la clave de dicho usuario, como se observa en la *ilustración 15.*

Ilustración 16.Diccionario2 SQLPing v3.0.

El ataque se genera nuevamente como se observa en la *Ilustración 16*.

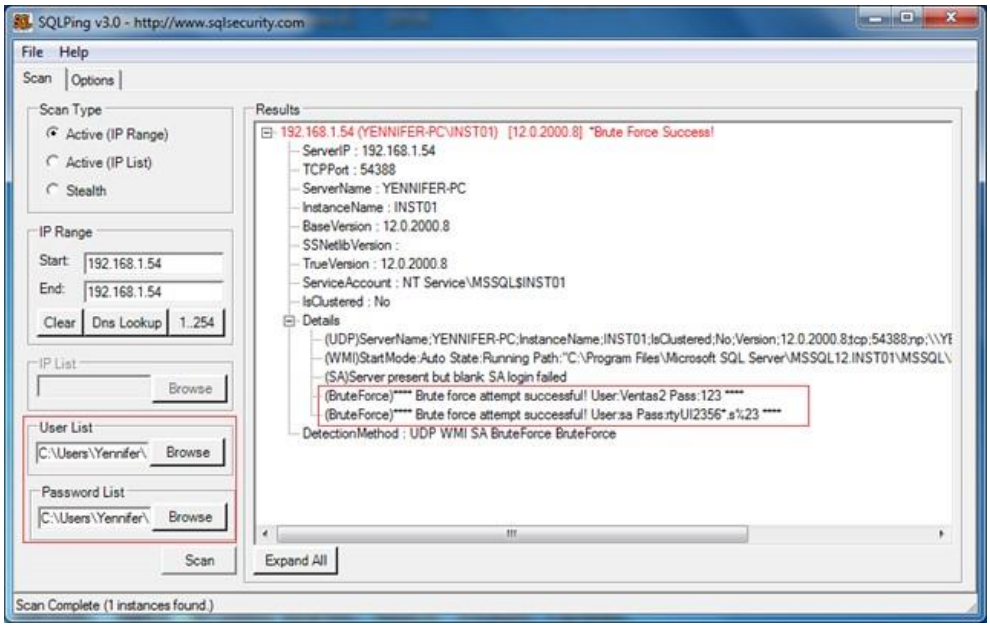

Ilustración 17.Ataque de diccionario2 SQLPing v3.0.

El resultado de este ataque es exitoso, SQLPing v3.0. logro encontrar la clave del usuario "sa". (Quezada, 2016)

- Advanced SQL Password Recovery UNREG

Esta herramienta logra acceder al servidor SQL Server, identifica los usuarios creados y permite cambiar la contraseña que tienen asignada, sin embargo, esta herramienta tiene un costo y por ello solo se utilizó para identificar los usuarios.

La herramienta solicita acceso al archivo master.mdf y con este trata de obtener el listado de los usuarios, como se muestra en la *Ilustración 17*.

*Ilustración 18.*Advanced SQL Password1

La herramienta no muestra la contraseña que el usuario tiene asignada, pero sí permite reemplazarla por una nueva, y para realizarlo solicita el pago, como se ve en la I*lustración 18.*

*Ilustración 19.*Advanced SQL Password2

- Kali Linux

Es un sistema operativo basado en Debian GNU/Linux diseñado principalmente para la auditoría y seguridad informática en general, cuenta con diferentes aplicaciones preinstaladas para poner a prueba la seguridad de los sistemas.

Para la experimentación se realizó la instalación de Kali Linux en una máquina virtual (Virtual Box) y se utilizaron las siguientes aplicaciones:

- Crunch

  Es un generador de listas de palabras donde se puede especificar la longitud y el conjunto de caracteres para que se cree un archivo con todas las combinaciones y permutaciones posibles.

  La creación de un diccionario se ejecuta mediante un comando que contiene la cantidad mínima y máxima de caracteres, las opciones, pueden ser letras, números y caracteres especiales, y por último el nombre del archivo (Crunch <min> <max> [opciones] > nombre_archivo.txt), por ejemplo:
  Crunch 3 4 123abc* > diccionario.txt

  Para crear un diccionario que contenga las claves de los usuarios registrados en SQL Server, es necesario que las listas de palabras tengan una longitud de 15 caracteres, con combinaciones de letras, números y caracteres especiales, además se requiere que la maquina en la que corre Kali Linux cuente con muy buena capacidad de procesamiento para que la aplicación sea capaz de crear el diccionario con las especificaciones mencionadas.

En la *Ilustración 19.* Se observa la ejecución del diccionario.

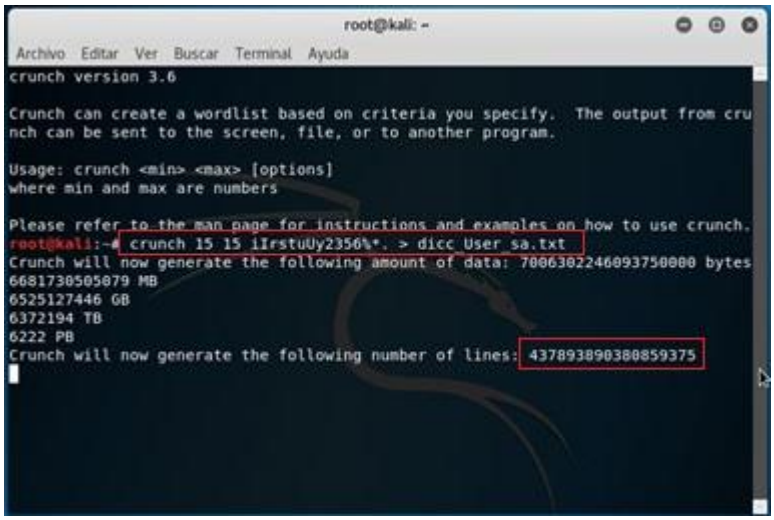

*Ilustración 20.*Diccionario Crunch SQL Server

Teniendo en cuenta las opciones ingresadas, el archivo tendría 437.893.890.380.859.375 líneas, por lo tanto, el resultado obtenido no es el esperado, debido a que el proceso se ejecuta por mucho tiempo y no logra terminar (más de 10 horas), es necesario matar el proceso para evitar que la maquina se bloquee.

Este resultado posiblemente se debe a que la máquina virtual solo permite utilizar 4 procesadores de los 8 disponibles en el pc y 7 gigas de RAM de las 12 disponibles.

- HexorBase: es una aplicación de base de datos diseñada para administrar y auditar múltiples servidores de bases de datos simultáneamente desde una ubicación centralizada, es capaz de realizar consultas SQL y ataques de fuerza bruta contra servidores de bases de datos como: MySQL, SQLite, Microsoft SQL Server, Oracle y PostgreSQL. (Ekiko, 2014)

Con esta aplicación se realiza un ataque de diccionario, especificando la dirección IP del servidor, el puerto y los mismos diccionarios de Usuarios y Claves observados en la *Ilustración 15. Diccionario2 SQLPing v3.0.*

En la *Ilustración 20*. Se puede ver el resultado del ataque

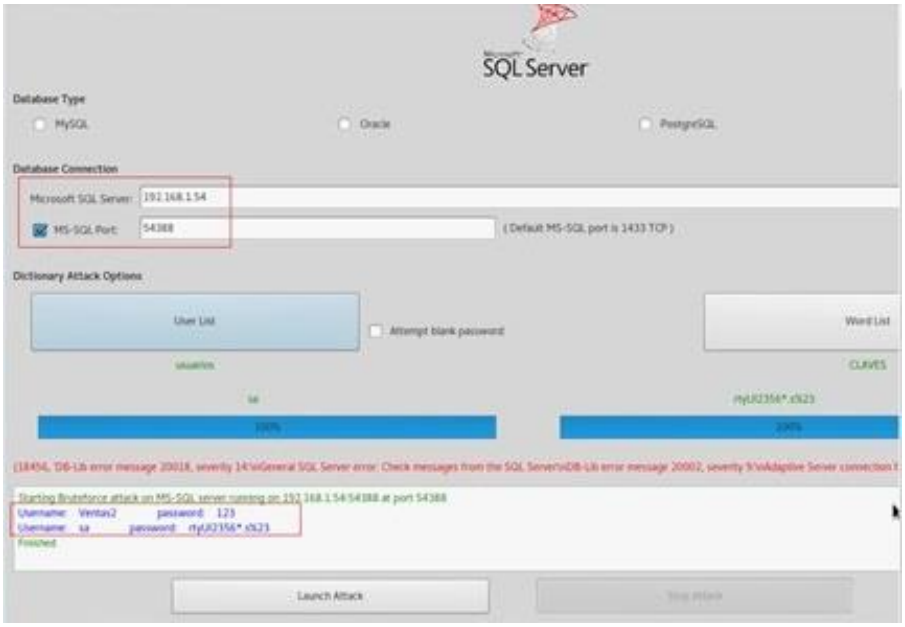

Ilustración 21.Ataque HexorBase SQL Server

El resultado es exitoso debido a que se utilizó el diccionario que contiene las claves reales de los usuarios.

- Johnny Es una aplicación con interfaz gráfica que desencripta hashes de contraseñas. (Shinnok, 2014)

  Para el ataque con esta aplicación se creó un archivo de texto con dos registros, que corresponden a los hashes de las contraseñas de los usuarios “sa” y “Ventas2”, las cuales son almacenadas en la vista sys.sql_login de SQL Server.

En la *Ilustración 21* se observa el resultado de la vista sys.sql_login con los hashes de las contraseñas.

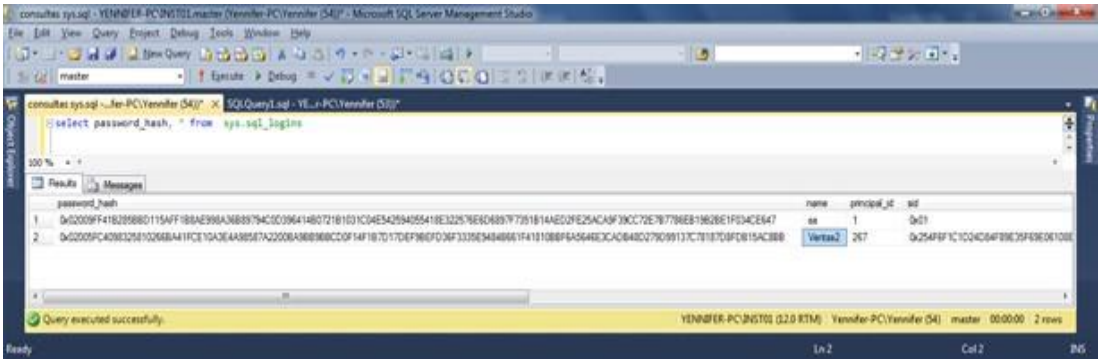

Ilustración 22.Vista sys.sql_login

En la *Ilustración 22* se muestra el archivo de texto a utilizar en el ataque.

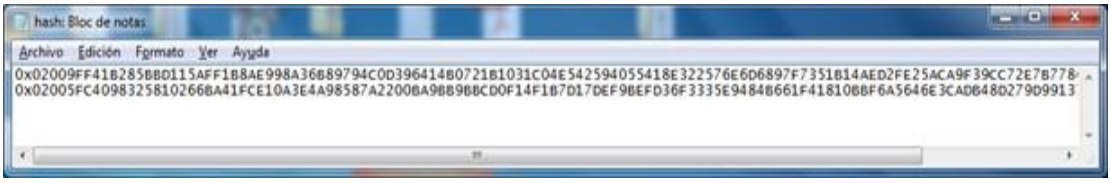

Ilustración 23.Archivo texto Johnny SQL Server.

En la aplicación Johnny se carga el archivo de texto y se inicia el ataque seleccionando la opción “Start new attack”.

En la *Ilustración 23* se visualiza el resultado del ataque.

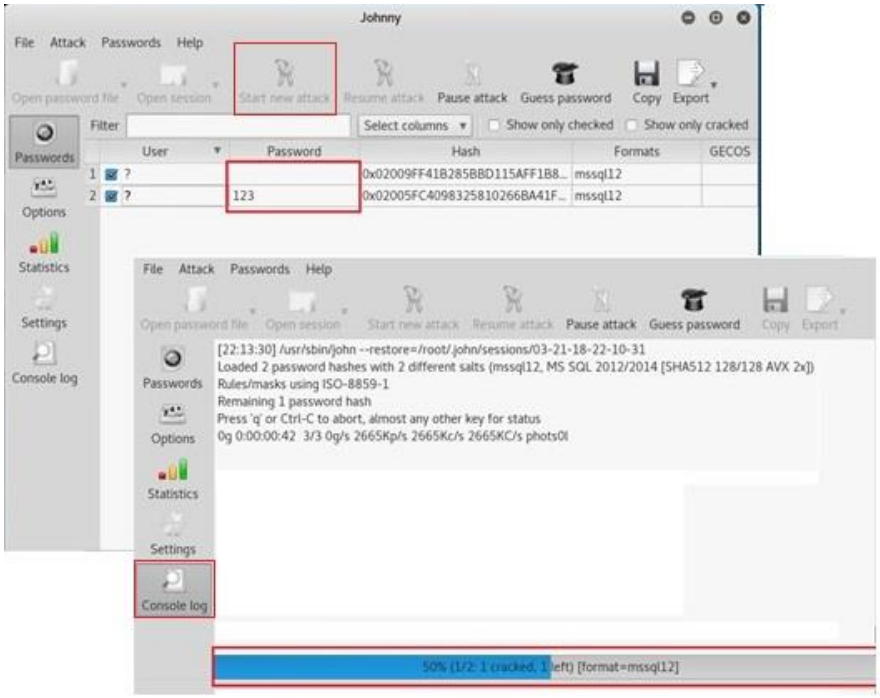

*Ilustración 24.*Ataque Johnny SQL Server

El resultado después de aproximadamente 16 horas solo es del 50%, se logra desencriptar la contraseña "123" del usuario "Ventas2", la contraseña para el usuario "sa" requiere de mayor capacidad de procesamiento, ya que tiene 15 caracteres de longitud, con combinación de letras (mayúsculas y minúsculas) números y caracteres especiales.

NOTA: también se realizaron pruebas con las herramientas Cain & Abel, Metasploit, NGSSQuirreL For SQL Server_Eval y wireshark estas pruebas no tuvieron éxito para el motor SQL Server.

- Ataque utilizando SQL Server en modo de usuario Único

Se realizó un ataque suponiendo que el intruso es un usuario interno que tiene acceso a la información de la red, el servidor y la instancia. utilizando el modo de usuario único

de SQL Server, el cual permite que se conecte al servidor únicamente un usuario a la vez.

Para este caso se pretende iniciar SQL Server en modo de usuario único para cambiar la contraseña del usuario sa.

Los pasos a seguir son los siguientes:

1. Ingresar a la línea de comandos cmd.
2. Detener la instancia de SQL Server con el comando: NET STOP MSSQL$INST01
3. Iniciar la instancia en modo de usuario único con el comando: NET START MSSQL$INST01 /m.
4. Iniciar la utilidad sqlcmd (esta permite escribir instrucciones Transact-SQL) con el comando SQLCMD -S seguido del nombre del servidor y la instancia, por ejemplo: SQLCMD -S YENNIFER-PC\INST01
5. Ingresar en la primera línea la instrucción Transact-SQL para modificar el Password del usuario sa, en la segunda línea la instrucción “go” y finalizar con “exit”. El comando es el siguiente:

```
1> Alter Login sa with Password=’Abcd1234.*’;
2> go
3> exit
```

6. Detener de nuevo la instancia:
7. Iniciar el servicio como multiusuario.

En la *ilustración 24* se muestran todos los pasos anteriores.

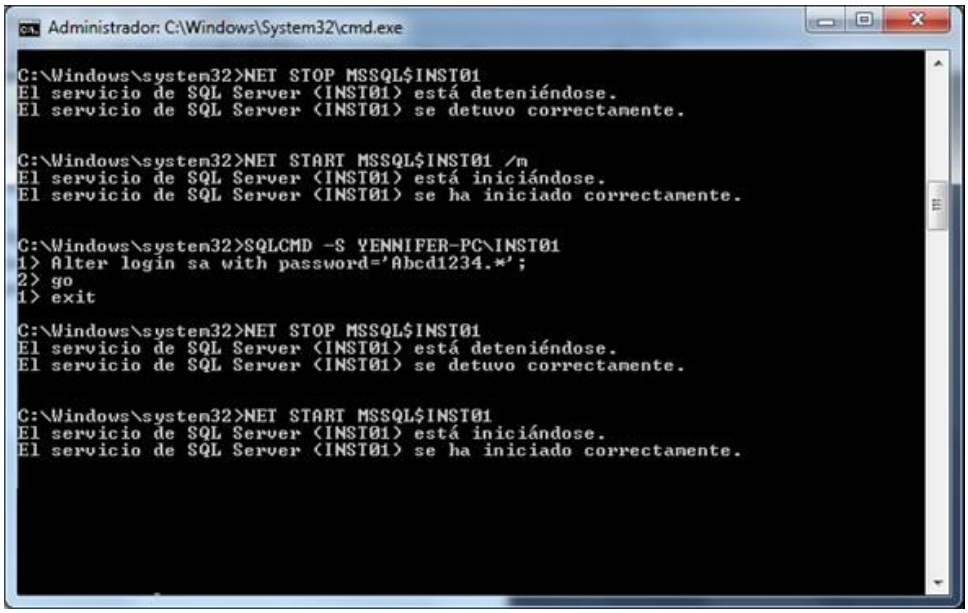

*Ilustración 25.*SQL Server modo de usuario Único

Finalizado el proceso anterior se ingresa a "SQL Server 2014 Management Studio", y se realiza la conexión con la autenticación de SQL con el usuario sa y la clave anteriormente asignada.

- Ataques Cifrado de datos para este proceso en el motor SQL Server se desarrolló un script que permite descifrar los datos cifrados de cada tabla (sales.CreditCard, HumanResources.EmployeePayHistory, Production.Product), teniendo en cuenta que el proceso de cifrado jerárquico que se implementó consta de varias capas, es necesario descubrir cada capa hasta llegar a los datos para descifrarlos, para ello el algoritmo realiza los siguientes pasos:

- Abrir la clave simétrica con la cual se protegen los datos, que a su vez está protegida con un certificado, se realizan todas las posibles combinaciones con claves simétricas y certificados, buscando con cuál de ellas se lograr abrir.
- Encontrar la función de cifrado, la tabla, la columna y comprobar si la clave simétrica que se encuentra abierta permite descifrar los datos, de lo contrario realiza la búsqueda de nuevo hasta que dé con un resultado positivo.
- Inicia de nuevo el proceso con una nueva combinación de clave y certificado hasta que logra descifrar los datos de todas las tablas ingresadas.

  A continuación, se encuentra el script desarrollado por los integrantes del equipo del trabajo de grado.

```
“Algoritmo para descifrar los datos
Use AdventureWorks2014
go
Declare @contK int;
Declare @contC int ;
Declare @Clave varchar (50);
Declare @Certificado varchar (50);
Declare @Val int =0;
Declare @ban bit =1;
Declare @ban2 bit =1;
Declare @ContM int =1;
Declare @ContT int =1;
Declare @ContT2 int =1;
Declare @Value nvarchar (500);
Declare @Value2 nvarchar (500);
Declare @Value3 nvarchar (500);
Declare @max int;
Declare @max2 int;
declare @Sql nvarchar (500);
declare @Sql2 nvarchar (500);
declare @Sql3 nvarchar (500);
Declare @Consulta1 varchar (50);
Declare @Consulta2 varchar (50);
Declare @Metodo1 varchar (50);
declare @tabla nvarchar (50);
declare @campoencryptado nvarchar (50);

--Se inicializan los contadores para las claves simetricas y los certificados
set @contK=(select min(symmetric_key_id) from sys.symmetric_keys where name not
like '##%')
```

```
set @contC=(select min(certificate_id) from sys.certificates)

--Se crean la tabla para ingresar los metodos y se insertan los valores
Create table #tblMetodos
( id int,
  NombreEncryp nvarchar (80),
  NombreDecryp nvarchar (80)
);
insert into #tblMetodos values ( 1,'ENCRYPTBYKEY','DECRYPTBYKEY');
insert into #tblMetodos values ( 2,'ENCRYPTBYPASSPHRASE','DECRYPTBYPASSPHRASE');

--Se crea la tabla para ingresar el nombre de la tabla y la columna cifrada y se
ingresan los valores
Create table #tblTablasEncryp
( id int,
  NombreTablaEncryp nvarchar (80),
  NombreCampoEncryp nvarchar (80)
);
insert into #tblTablasEncryp values (
1,'HumanResources.EmployeePayHistory','EncryptedRate');
insert into #tblTablasEncryp values (
2,'Sales.CreditCard','EncryptedCardNumber');
insert into #tblTablasEncryp values (
3,'Production.Product','EncryptedStandardCost');

--Se inicializa el contador del ciclo principal
set @max2= (select max(symmetric_key_id) from sys.symmetric_keys)

WHILE @contK<= @max2
BEGIN
  --Se obtiene el nombre de la clave simetrica
  set @Clave=(select name FROM sys.symmetric_keys where symmetric_key_id=@contK)
  --Se obtiene el nombre del certificado
  set @Certificado=(select name FROM sys.certificates where
certificate_id=@contC)

  --Se abre la clave simetrica con el certificado
  set @Sql=N'Open symmetric key ' + @Clave + ' decryption by certificate ' +
@Certificado
  execute sp_executesql @Sql;

  --se comprueba si efectivamente se abrió la clave simétrica y se guarda el
resultado en @Value3
  set @Value=N'select @Value2=key_id from sys.openkeys where key_id= ' +
CONVERT(nvarchar (50),@contK)
  execute sp_executesql @Value, N'@Value2 nvarchar (50) OUTPUT', @Value3 OUTPUT

  --Si @Value3 es null se aumenta el contador de los certificados y se vuelve a
ingresar al while, de lo contrario inicia otro while
  --para descifrar la columna de la tabla con la clave simétrica abierta
  IF (@Value3) IS NOT NULL
   BEGIN
     --Se inicializa los contadores para el otro while y para las tablas cifradas
     set @max= (select max(id) from #tblTablasEncryp)
     set @ContM=1
```

```
     set @ContT2=1
      WHILE @ContM<=@max
       BEGIN
        --Se obtiene el nombre del método, la tabla y la columna
        set @Metodo1 =(select NombreDecryp from #tblMetodos where id=@ContT2)
        set @tabla =(select NombreTablaEncryp from #tblTablasEncryp where
id=@ContT)
        set @campoencryptado =(select NombreCampoEncryp from #tblTablasEncryp
where id=@ContT)

        --se comprueba si la clave simétrica abierta logra descifrar los datos de
la columna y se guarda el resultado en @Consulta2
        Set @Sql2=N'select top 1 @Consulta1=CONVERT(nvarchar, ' +@Metodo1+
'('+@campoencryptado+ '))
        from '+ @tabla
        execute sp_executesql @Sql2, N'@Consulta1 nvarchar (50) OUTPUT',
@Consulta2 OUTPUT

        --Si @Consulta2 no es Null muestra la columna descifrada y los demás
campos de la tabla,
        --de lo contrario comprueba el descifrado con las demás tablas y métodos
        IF @Consulta2 IS NOT NULL
         BEGIN
           Set @Sql2=N'select CONVERT(nvarchar, ' +@Metodo1+
'('+@campoencryptado+ ')) AS ''Valor desencriptado'',*
           from '+ @tabla
           execute sp_executesql @Sql2, N'@Consulta1 nvarchar (50) OUTPUT',
@Consulta2 OUTPUT

           --Se reinician los contadores y se cierra la clave simétrica abierta
           set @ContM=@max+1
           set @Consulta2=NULL
           Set @Sql3=N'close symmetric key '+@Clave
           execute sp_executesql @Sql3
           -- en la variable @Val se cuenta cada columna descifrada
           set @Val=@Val+1
         END
        ELSE
         BEGIN
           --Validación para finalizar cuando la cantidad de columnas descifradas
sea igual a la cantidad de tablas registradas
           IF @Val>=@max
            BEGIN
              SET @ContM=@max+1
              SET @contK=@max2+1
            END
           ELSE
            BEGIN
            --Validación para iniciar el ciclo de nuevo con otro método
              IF @ContT=@max
               BEGIN
                 set @ContT2=@ContT2+1
                 set @ContT=1
                 set @ContM=1
               END”
```

```
                    ELSE
                    BEGIN
                      set @ContM=@ContM+1
                      set @ContT=@ContT+1
                    END
                  END
                END
              END
              SET @contK=@contK+1
              SET @contC=(select min(certificate_id) from sys.certificates)
              SET @Value3=NULL
              SET @ContT=1
          END
          ELSE
           BEGIN
             set @contC=@contC+1
           END
       END;
Construcción: Yennifer Villa, 2018
```

Resultado de la ejecución del script:

Las columnas con la información cifrada requerida para la simulación del ataque fueron descifradas de manera satisfactoria, obteniendo así los datos en texto plano listos para su utilización.

3.2.1.2 Métodos de seguridad MySQL

A. Caso práctico control de acceso se realiza la instalación típica de MySQL versión 5.7 y se instala el cliente (GUI) Workbench versión 6.3. En cuanto a seguridad durante la instalación se tiene en cuenta que la clave del usuario "root" (administrador del sistema) cumpla con las características de una contraseña segura.

Se utiliza la base de datos "Adventureworks2014".

En la *Ilustración 25* se observa en que parte de la instalación se asigna la clave segura para el usuario "root".

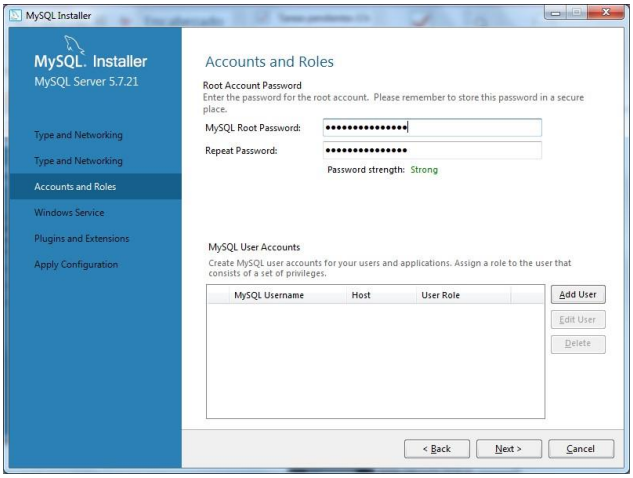

*Ilustración 26.*Password usuario root.

- Implementación sistema de privilegios de acceso para la implementación del sistema de privilegios es necesario ingresar a cada tabla la información de los usuarios y sus respectivos permisos.

  La definición de los usuarios y privilegios a utilizar se encuentran en la *Tabla 1. Roles, permisos y usuarios.*

  Tabla user: en esta tabla se insertan los usuarios que van a acceder al servidor de MySQL, la descripción de las columnas se menciona a continuación:

  - Host: el equipo o pc del usuario
  - User: nombre con el que se identifica el usuario.
  - Plugin: método de cifrado de la contraseña.
  - authentication_string: la contraseña asignada al usuario.

  Las demás columnas quedan por defecto con el valor ‘N’ (indica acceso denegado).

En la *Tabla 4* se muestra el contenido de la tabla user.

Tabla 4.Contenido tabla user MySQL

Tabla user			
Host	user	plugin	authentication_string
localhost	Root	mysql_native_password	rtyUI2356*.s%23
localhost	Ventas1	mysql_native_password	fdo6783%.#e3
localhost	ConfigVentas1	mysql_native_password	eri0529,$(i9
localhost	Cumplimiento1	mysql_native_password	qwcs1830&#.v4
localhost	Inventario1	mysql_native_password	dfli2571)%,d6
localhost	Compras1	mysql_native_password	awty1045%$.h7
localhost	Recurso_humano1	mysql_native_password	cvmn9857#%)o1
localhost	Presupuestos1	mysql_native_password	iozx7239)$/b3
localhost	Produccion1	mysql_native_password	fwph610·&.w2
localhost	Consultas1	mysql_native_password	huip(#45yop.

Se utiliza la siguiente instrucción sql para insertar cada uno de los usuarios en la tabla:

CREATE USER 'Nombre_Usuario'@'localhost' IDENTIFIED BY 'Password_Usuario';

En la *Ilustración 26* se observan los usuarios creados.

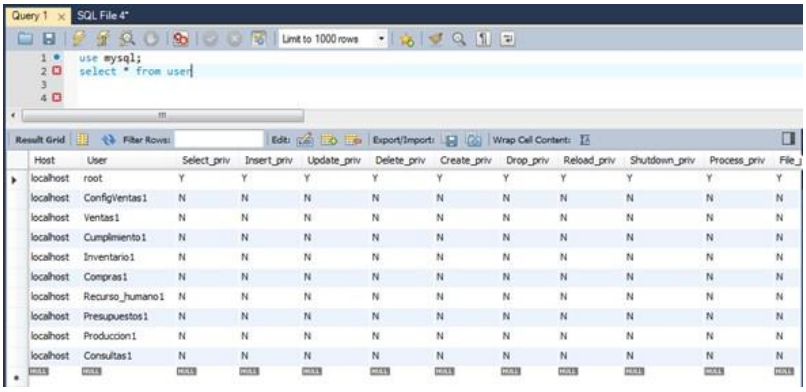

*Ilustración 27.*Usuarios de la tabla user MySQL.

Tabla db: en esta tabla se insertan los mismos usuarios de la tabla user y los permisos que van a tener sobre la base de datos adventureworks.

La descripción de las columnas se menciona a continuación:

- Host: el equipo o pc del usuario.
- db: nombre de la base de datos a la que se asigna el acceso.
- User: nombre con el que se identifica el usuario.
- Select_priv: permiso para seleccionar (posibles valores 'Y'(si) o 'N' (no)).
- Update_priv: permisos para actualizar o modificar (posibles valores 'Y'(si) o 'N' (no)).

Las demás columnas quedan por defecto con el valor 'N' (indica acceso denegado)

En la *Tabla 5* se muestra el contenido de la tabla db.

*Tabla 5.*Contenido tabla db MySQL.

Tabla db				
Host	Db	User	Select_priv	Update_Priv
localhost	adventureworks	Ventas1	Y	N

localhost	adventureworks	ConfigVentas1	Y	Y
localhost	adventureworks	Cumplimiento1	Y	N
localhost	adventureworks	Inventario1	Y	Y
localhost	adventureworks	Compras1	Y	Y
localhost	adventureworks	Recurso_humano1	Y	Y
localhost	adventureworks	Presupuestos1	Y	Y
localhost	adventureworks	Produccion1	Y	Y
localhost	adventureworks	Consultas1	Y	N

Se utiliza la siguiente instrucción sql para insertar cada uno de los datos en la tabla:

Insert into db (Host, Db, User, Select_priv, Update_priv) values

('Host','Nombre_BD', 'Nombre_Usuario',' Select_priv ',' Update_priv');

En la *Ilustración 27* se visualizan los permisos de cada usuario en la base datos “adventureworks”.

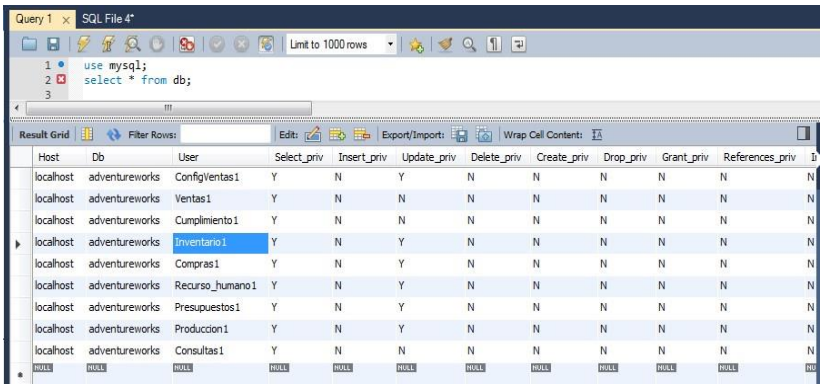

*Ilustración 28.*Permisos tabla db MySQL.

Tabla tables_priv: se insertan los nombres en las tablas de la BD “adventureworks” a las que va a tener acceso cada usuario.

La descripción de las columnas se menciona a continuación:

- Host: el equipo o pc del usuario.
- Db: nombre de la base de datos a la que se asigna el acceso.
- User: nombre con el que se identifica el usuario.
- table_name: nombre de la tabla de la base de datos
- table_priv: privilegio asignado (posibles valores 'Select', 'Insert', 'Update', 'Delete', 'Create', 'Drop', 'Grant', 'References', 'Index', 'Alter', 'Create View', 'Show view', 'Trigger).
- Column_priv: se deja en blanco indicando que no se asigna ningún permiso a nivel de columna.

En la *Tabla 6* se muestra el contenido de la tabla tables_priv.

Tabla 6. Contenido tabla tables_priv MySQL.

Tabla tables_priv				
Host	db	user	table_name	table_priv
localhost	adventureworks	Ventas1	CountryRegionCurrency, CreditCard, Currency, CurrencyRate,Customer,PersonCreditCard,SalesOrderDetail,SalesOrderHeader,SalesOrderHeaderSalesReason,SalesPerson,SalesPersonQuotalHistory,SalesReason,SalesTaxRate,SalesTerritory,SalesTerritoryHistory,ShoppingCardItem,SpecialOffer, SpecialOfferProduct,Store	select
localhost	adventureworks	Config Ventas1	CountryRegionCurrency, CreditCard, Currency, CurrencyRate,Customer,PersonCreditCard,SalesOrderDetail,SalesOrderHeader,SalesOrderHeaderSalesReason,SalesPerson,SalesPersonQuotalHistory,SalesReason,SalesTaxRate,SalesTerritory,SalesTerritoryHistory,ShoppingCardItem,SpecialOffer, SpecialOfferProduct,Store	select, update
localhost	adventureworks	Cumplimiento1	WorkOrder, ScrapReason, ShoppingCartItem	select
localhost	adventureworks	Inventario1	BillOfMaterials,Culture,Document,Illustration,Location,Product,ProductCategory,ProductCostHistory,ProductDescription,ProductDocument,ProductI	select, update

			nventory,ProductListPriceHistory,ProductModel, ProductModelIllustration,ProductModelProductDescriptionCulture,ProductPhoto,ProductProductPhoto,ProductReview,ProductSubcategory,ScrapReason,TransactionHistory,TransactionHistoryArchive,UnitMeasure,WorkOrder,WorkOrderRouting	
localhost	adventureworks	Compras1	ProductVendor,PurchaseOrderDetail,PurchaseOrderHeader,ShipMethod,Vendor,Address,Contact	select, update
localhost	adventureworks	Recurso_humano1	Department,Employee,EmployeeDepartmentHistory,EmployeePayHistory,JobCandidate,Shift	select, update
localhost	adventureworks	Presupuestos1	Currency,SpecialOffer, SpecialOfferProduct	select, update
localhost	adventureworks	Produccion1	BillOfMaterials,Document,Illustration,Location,Product,ProductInventory,ProductModel,ScrapReason,WorkOrder,WorkOrderRouting	select, update
localhost	adventureworks	Consultas1	Store , Product, ShoppingCartItem, SpecialOfferProduct,SpecialOffer	select

Se utiliza la siguiente instrucción sql para insertar cada uno de los datos en la tabla:

Insert into tables_priv (Host, Db, User, Table_name, Table_priv) values

('Host','Nombre_BD', 'Nombre_Usuario',' table_name ',' table_priv ');

En la *Ilustración 28* se visualizan los privilegios de cada usuario en las tablas de la BD "adventureworks".

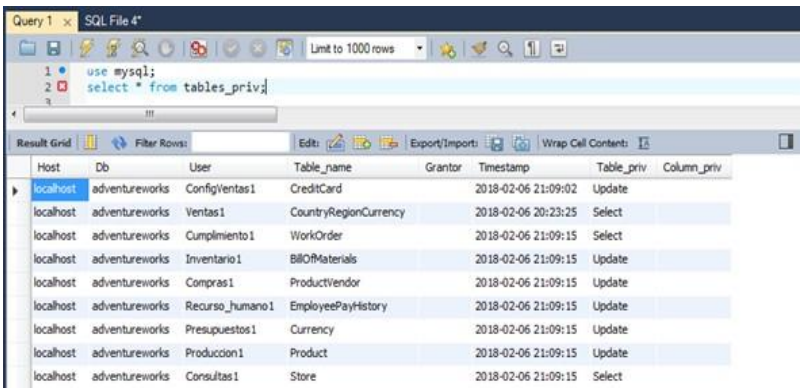

Ilustración 29.Privilegios tabla tables_priv MySQL.

B. Caso práctico cifrado de datos: para el proceso de cifrado en MySQL se utilizaron las tablas y columnas definidas en la *Tabla 2. Tablas y columnas cifradas de datos*

Pasos para realizar el cifrado de datos

1. Modificar tabla para agregar la columna que va a contener los datos cifrados

   Instrucción Sql: alter table Nombre_Tabla add Nombre_Columna_Nueva varbinary (128);

2. Cifrar los datos de la columna utilizando la función de cifrado y agregarlos a la nueva columna creada.

   Instrucción Sql funciones MD5 y SHA1: Update Nombre_Tabla set Nombre_Columna_Nueva = Funcion_Cifrado (Nombre_Columna_a_Cifrar);

   Instrucción Sql función AES_ ENCRYPT: Update Nombre_Tabla set Nombre_Columna_Nueva = Funcion_Cifrado (Nombre_Columna_a_Cifrar, 'Clave_de_Cifrado');

Nota: en cada tabla se utilizó una función de cifrado diferente.

En la *Ilustración 29.* se observa la columna con los datos cifrados después de ejecutar las instrucciones Sql.

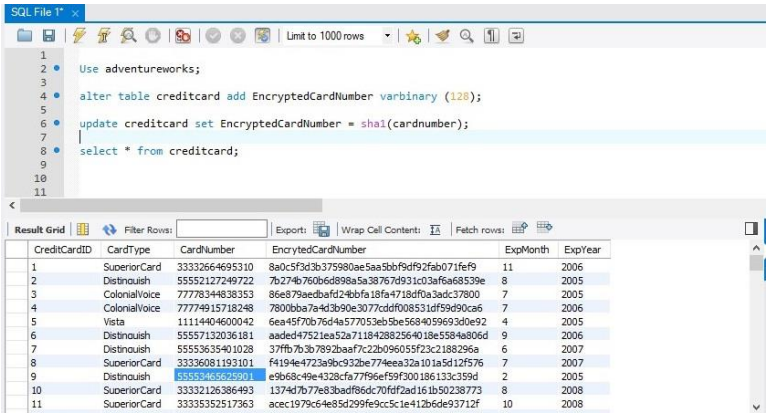

*Ilustración 30.*Datos cifrados MySQL.

C. Ataques: los ataques en MySQL se realizaron al igual que en SQL Server, mediante un test de intrusión de caja gris, para encontrar las falencias de seguridad, en el control de acceso (contraseñas de usuarios) y en los datos cifrados en la base de datos "adventureworks".

- Ataques control de acceso para los ataques en MySQL se utilizaron los usuarios "root" y "Ventas1", se asignó una clave segura y una no segura a cada usuario, como se muestra en la *Tabla 7.*

*Tabla 7.*Usuarios MySQL.

Usuarios	**Claves**
root	rtyUI2356*.s%23
Ventas1	123

Para el proceso se utilizaron las herramientas Cain & Abel, HexorBAse y Johnny, como se detalla a continuación:

- HexorBase: con esta aplicación se realiza un ataque de diccionario, especificando la dirección IP del servidor, el puerto y los diccionarios de Usuarios y Claves que se observan en la *Ilustración 30.*

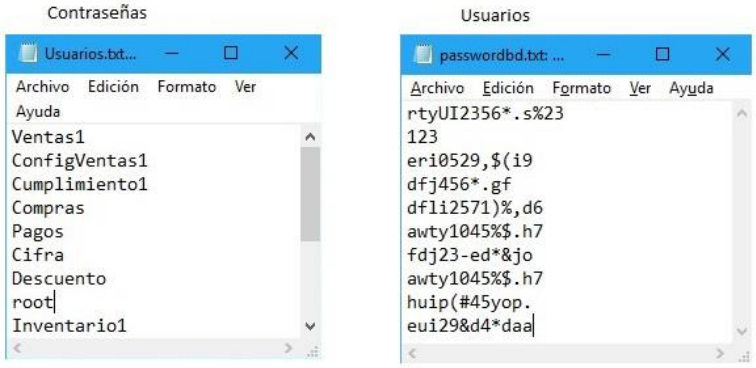

Ilustración 31.Diccionarios HexorBase MySQL

El resultado del ataque se visualiza en la *Ilustración 31*.

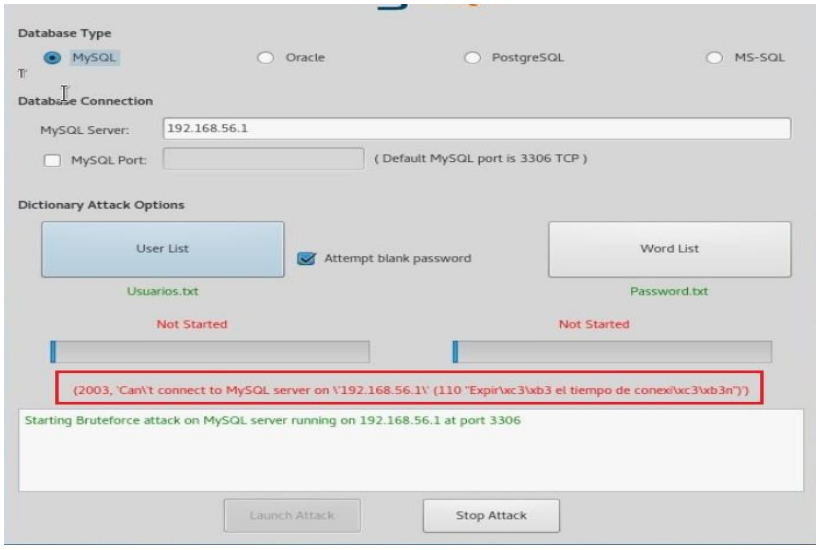

Ilustración 32.Ataque1 HexorBase MySQL

En el resultado se puede evidenciar que el ataque no fue exitoso, esto se debe a que para acceder al servidor de MySQL se necesita, el usuario, la contraseña y el host que se encuentra almacenado en la tabla user y HexorBase se intenta conectar con el usuario y la contraseña correctos, pero con un host diferente al definido en la tabla user.

En este caso para realizar un ataque exitoso sería necesario modificar la configuración del acceso al servidor de MySQL para que permita la conexión desde cualquier Host, esto se puede hacer utilizando un comodín (%) en la columna Host.

En la *Ilustración 32* se observa la modificación del usuario "root".

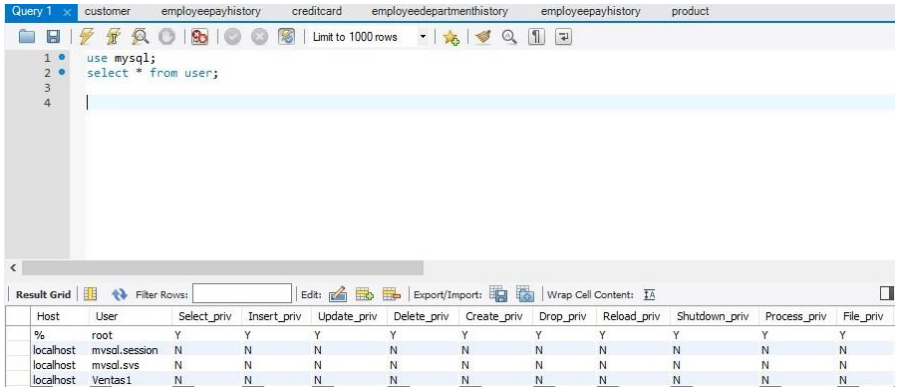

Ilustración 33.Usuario "root" modificado MySQL.

Al generar el ataque nuevamente con HexorBase, después de modificar el usuario "root", el resultado es exitoso, como se visualiza en la *Ilustración 33*

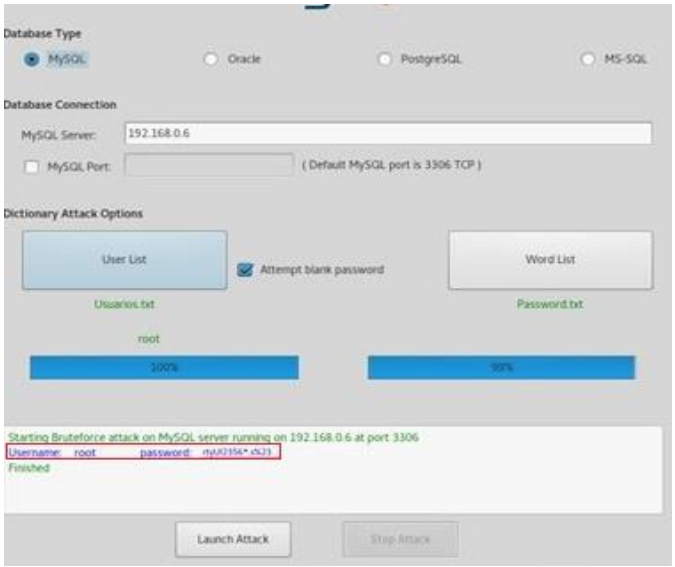

Ilustración 34.Ataque2 HexorBase MySQL

- Johnny: para el ataque con esta aplicación se creó un archivo de texto con dos registros, que corresponden a los hashes de las contraseñas de los usuarios "root" y "Ventas1", los cuales se almacenan en la tabla user de MySQL.

En la *Ilustración 34* se encuentra el resultado del ataque.

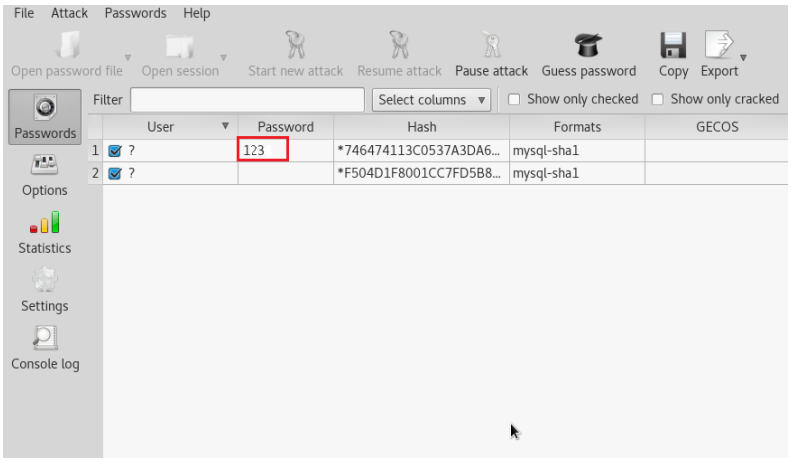

Ilustración 35.Ataque Johnny MySQL

En el resultado se observa después de aproximadamente 16 horas que Johnny logra descifrar la contraseña "123" del usuario Ventas1, mientras que la contraseña del usuario "root" no se logra descifrar, este resultado es igual al obtenido en el ataque de SQL Server con esta misma herramienta, ya que la contraseña del usuario "sa" y del "root" tienen la misma complejidad.

- Ataques cifrados de datos: para esta segunda parte se pretende descifrar la información de las columnas cifradas de las tablas: EmployeePayHistory, CreditCard y Product.

  Se utilizaron las herramientas Cain & Abel y Johnny.

  - Cain & Abel: es una herramienta de recuperación de contraseña para los sistemas

operativos de Microsoft. Permite recuperar fácilmente varios tipos de contraseñas, descifrando contraseñas cifradas usando técnicas de diccionario, fuerza bruta y criptoanálisis, su objetivo principal es la recuperación simplificada de contraseñas y credenciales de diversas fuentes. (Montoro, 2014)

Para el ataque se utiliza el valor cifrado de uno de los registros almacenados en la tabla EmployeePayHistory (956aac64bc5911c4c320e7501a02de3d), esta tabla se cifro con la función MD5 y por tanto en Cain & Abel se selecciona la opción “MD5 Hashes”.

En la *Ilustración 35* se puede observar el resultado del ataque.

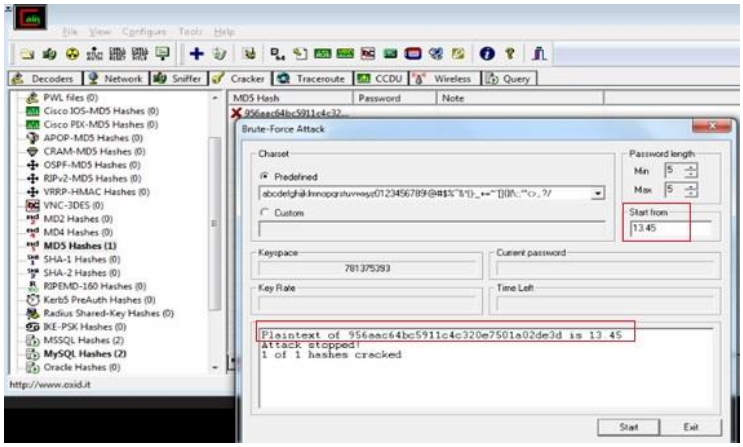

Ilustración 36.Ataque1 Cain & Abel MySQL.

El resultado de descifrar la cadena “956aac64bc5911c4c320e7501a02de3d” es el valor ”13.45”, lo cual es correcto, por lo tanto, el ataque fue exitoso.

Se ejecuta un nuevo ataque utilizando la opción “SHA-1 Hashes” para descifrar el número de la tarjeta de crédito almacenado en la tabla CreditCard, (7b274b760b6d898a5a38767d931c03af6a68539e)

En la *Ilustración 36* se observa el resultado obtenido del ataque.

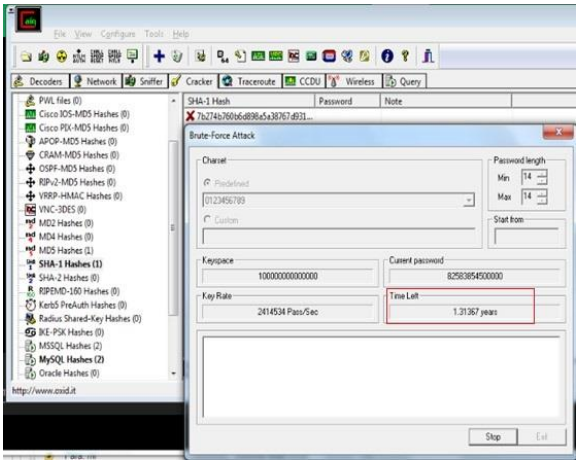

Ilustración 37.Ataque2 Cain & Abel MySQL.

Para este caso no se logra desencriptar el número de la tarjeta de crédito, dado que el valor contiene 14 caracteres de longitud y se requiere de mayor capacidad de procesamiento para descifrar una cadena de este tipo, la aplicación indica que faltan 1,31367 años para finalizar.

3.2.1.3 Métodos de seguridad cassandra.

A. Caso práctico control de acceso: se realiza la instalación típica de Cassandra la cual no incluye ninguna interfaz gráfica y se debe ejecutar por medio de la línea de comandos de Windows.

El proceso para la instalación es el siguiente:

1. Requisitos previos: Instalar la última versión de Java 8 y Python 2.7 y crear las variables de entorno para cada uno:

- Variables de usuario

- Nombre Variable: JAVA_HOME
- Valor de la variable: C:\Program Files\Java\jdk1.8.0_161

- Variables del sistema
  - Nombre Variable: Path
  - Valor de la variable: C:\Python27

2. Descargar la última versión de Apache Cassandra 3.11.2, desde el sitio oficial: http://cassandra.apache.org/download/
3. Descomprimir el archivo y pegar la carpeta “apache-cassandra-3.11.2” en la raíz del disco C y adicionar en la variable de entorno Path el valor:” C:\apache-cassandra-3.11.2\bin”.
4. En una ventana de comando de Windows ingresar el comando: > Cassandra.

En la *Ilustración 37* se puede observar que el proceso finaliza cuando la última línea indica que se creó por defecto el rol “cassandra” como super usuario.

Ilustración 38.Instalación Cassandra.

- Implementación control de acceso basado en Roles (RBAC). se realiza el método de control de acceso basado en Roles de Cassandra, el cual ya viene configurado por defecto

en el archivo cassandra.yaml, con el parámetro role_manager igual a CassandraRoleManager.

Esta configuración almacena la información de los roles y permisos en el keyspace "system_auth".

Una vez iniciado el servidor de Cassandra, el paso siguiente para la creación de roles es modificar el archivo de configuración "Cassandra.yaml" para que permita la autenticación y autorización de roles.

El archivo se encuentra en la ruta: "C:\apache-cassandra-3.11.2\conf" y se modifican los siguientes parámetros:

- Authenticator: se adiciona el valor PasswordAuthenticator, para que al iniciar Cassandra se solicite usuario y contraseña (Authenticator: PasswordAuthenticator).
- Authorizer: se adiciona el valor CassandraAuthorizer, para que se puedan configurar los permisos de cada rol (Authorizer: CassandraAuthorizer).

Una vez hecho lo anterior se debe reiniciar el host, volver a iniciar Cassandra y abrir otra ventana de comandos, para iniciar una sesión de "cqlsh" (Shell de línea de comandos de Cassandra, con el cual se pueden ejecutar instrucciones en el lenguaje de consultas CQL), con la autenticación del super usuario cassandra.

En la *Ilustración 38* se puede observar el ingreso a la utilidad "cqlsh" con la autenticación del super usuario cassandra.

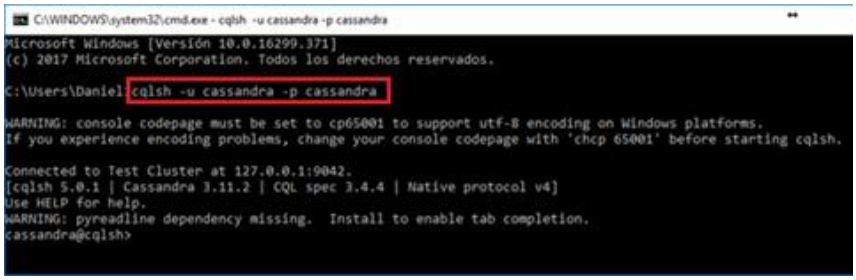

*Ilustración 39.*Ingreso a la utilidad "cqlsh" Cassandra.

A continuación, se procede con la creación del Keyspace (base de datos), de los roles y de los permisos:

- Keyspace: AdventureWorks2014.
- Tablas: salescreditcard, HumanResourcesEmployeePayHistory, ProductionProduct.
- Roles: Ventas, ConfigVentas, Inventario, Recurso_humano, Consultas.

Creación keyspace adventureworks2014: la base de datos se crea ejecutando la siguiente instrucción:

```
CREATE KEYSPACE adventureworks2014
WITH replication = {'class' : 'SimpleStrategy' , 'replication_factor' : 1 };
```

En la *Ilustración 39* se visualizan las bases de datos creadas, al ejecutar el comando "describe keyspaces".

*Ilustración 40.*Keyspace adventureworks2014 Cassandra.

- Creación de las tablas se procede solamente con la creación de tres tablas, ya que, cassandra es una base de datos NoSQL y por tanto no es posible restaurar la base de datos adventureworks2014 que maneja un modelo relacional.

Las tablas se crean ejecutando la siguiente instrucción:

CREATE TABLE Nombre_Tabla (Columna1 Tipo_Dato PRIMARY KEY, Columna2 Tipo_Dato, Columna3 Tipo_Dato…);

En la *Ilustración 40* se observan las tablas creadas, al ejecutar el comando "describe tables".

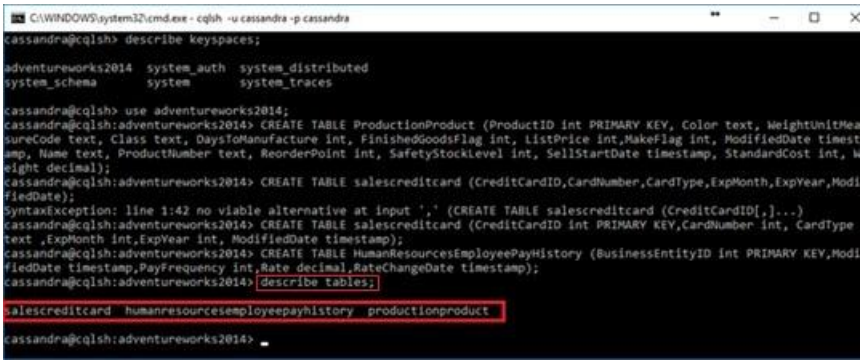

*Ilustración 41.*Tablas adventureworks2014 Cassandra.

- Creación de los roles y asignación de Permisos los roles y permisos se crean ejecutando las siguientes instrucciones:
  - Crear Rol

    CREATE ROLE Nombre_Rol WITH SUPERUSER = Valor_false_o_true AND LOGIN = Valor_ false_o_true AND PASSWORD = "Password_Usuario";

  - Asignación de permisos

    GRANT Privilegio ON Nombre_Keyspace.Nombre_Tabla TO Nombre_Rol;

  La información de los roles y sus permisos quedan almacenados en el keyspace "system_auth" en las siguientes tablas:

  - Roles: esta tabla contiene la información del nombre del rol, si tiene login habilitado, si es super usuario y la clave encriptada.
  - resource_role_permissons_index: la tabla contiene el rol, el keyspace y/o la tabla a la que tiene permisos el rol.
  - role_permissions: en esta tabla se encuentra el rol, el keyspace y/o la tabla y los permisos que tiene asignados el rol (select, update, insert…etc.)

  En la *Ilustración 41* se observan las tablas anteriores con la información ingresada.

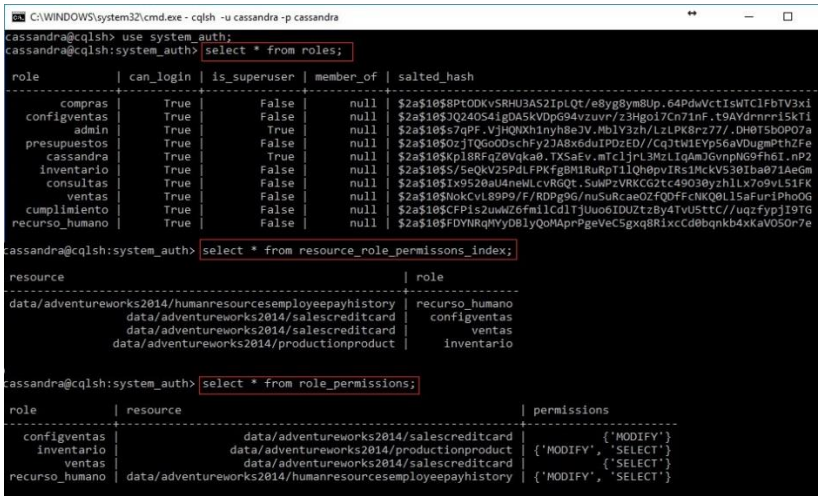

Ilustración 42.Tablas "system_auth" Cassandra.

En la *Ilustración 42*. se listan los permisos con lo que quedaron asignados los roles

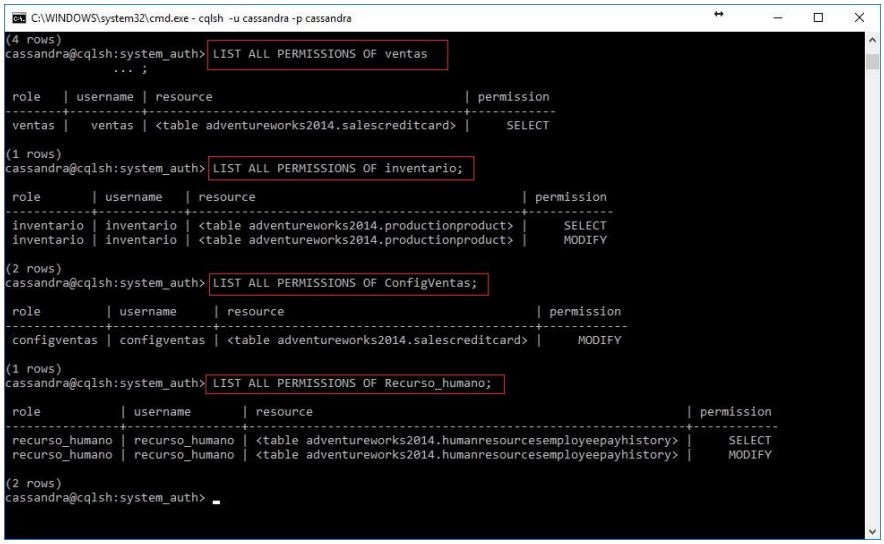

Ilustración 43.Permisos roles Cassandra.

B. Ataques: los ataques en Cassandra se realizaron al igual que en SQL Server y MySQL, mediante un test de intrusión de caja gris, para identificar falencias de seguridad, en el control de acceso (contraseñas de usuarios) en la base de datos "adventureworks".

- Ataques control de acceso para los ataques en Cassandra se utilizaron los usuarios "Cassandra" y "Ventas", se asignó una clave segura y una no segura a cada usuario, como se muestra en la *Tabla 8*.

*Tabla 8.*Usuarios Cassandra.

Usuarios	**Claves**
Cassandra	rtyUI2356*.s%23
Ventas	a1b2c3

Para el proceso se utilizó la herramienta Johnny de Kali Linux, como se detalla a continuación:

- Johnny Kali Linux: para el ataque con esta aplicación se creó un archivo de texto con dos registros, que corresponden a los hashes de las contraseñas de los roles, que son almacenados en la tabla "roles" del keyspace "system_auth".

En la *Ilustración 43* se observan los hashes de contraseña de los usuarios seleccionados para el ataque.

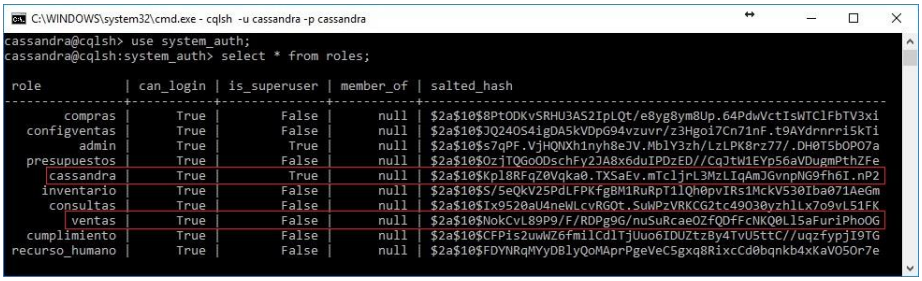

*Ilustración 44.*Hash Contraseñas Usuarios Cassandra.

En la *Ilustración 44* se visualiza el contenido del archivo de texto creado para el ataque.

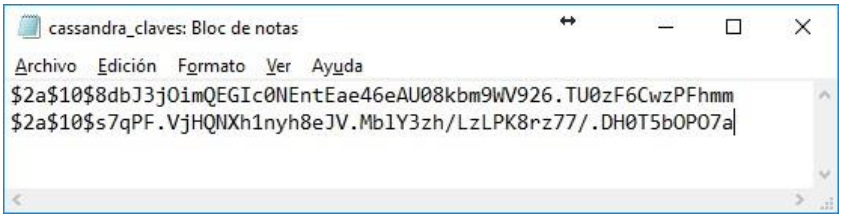

Ilustración 45.Archivo de texto Johnny Cassandra.

En la aplicación Johnny se carga el archivo de texto y se inicia el ataque.

En la *Ilustración 45* se muestra el resultado del ataque.

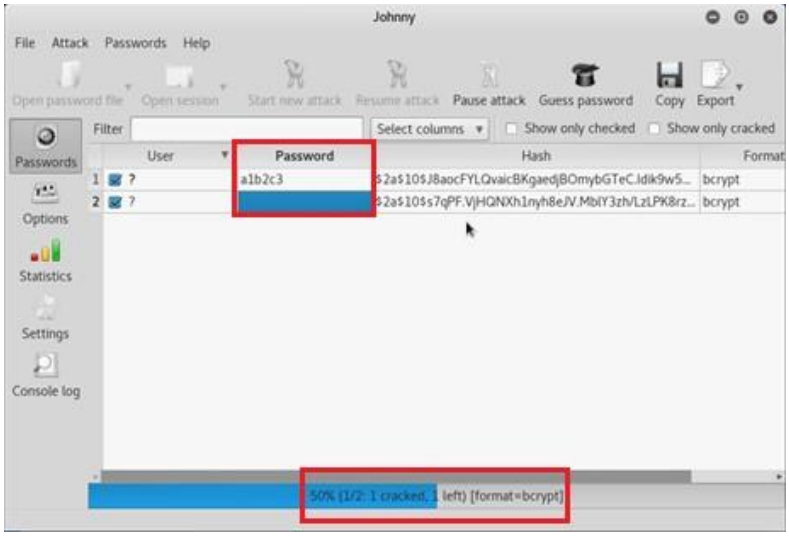

Ilustración 46.Ataque Johnny Cassandra.

Como era de esperarse el resultado obtenido es solo del 50%, se logra desencriptar la contraseña “a1b2c3” del rol ventas, la contraseña para el usuario Cassandra al igual que en SQL Server y MySQL requiere de mayor capacidad de procesamiento ya que tiene 15 caracteres de longitud.

3.3 Fase 3: Análisis: Con base en la información consultada en la fase 1 y en los resultados de la experimentación de la fase 2, se resaltaron en la Tabla 9 y Tabla 10 las principales ventajas y desventajas encontradas en cada uno de los motores y métodos de seguridad seleccionados.

3.3.1 Ventajas y desventajas motores de base de datos

*Tabla 9.*Ventajas y desventajas motores BD.

Motor	Ventajas	Desventajas
SQL Server	• Administraciónón de acceso a entidades de seguridad protegidas mediante autenticaciónón y autorizaciónón. • Seguridad basada en roles: permite la aplicaciónón de roles de servidor y de base de datos fijos y definidos por el usuario. • Dos tipos de autenticación: Autenticación de Windows: Autenticación SQL Server: • Reglas de seguridad a esquemas y a usuarios de forma independiente. • Permite el Cifrado de conexiones, datos y procedimientos almacenados. • Directivas de contraseñas que impiden ataques por fuerza bruta con el aumento de numero de contraseñas.	• Es más seguro cuando se instala en un controlador de dominio. • El inicio de sesión sa (system administrator) tiene permisos de administrador y credenciales administrativas irrevocables; si se asigna una contraseña no segura es posible que un atacante obtenga acceso fácilmente y pueda causar mucho daño. • SQL Server 2014 SP1 y SP2 permite a los usuarios remotos autenticados obtener privilegios a través de vectores desconocidos,esta vulnerabilidad se resuelve con la actualización de seguridad SQL Server 3199641. (Microsoft, 2016) • SQL Server por ser un motor más robusto genera un grado de complejidad mayor en el análisis e implementación de los procedimientos.
MySQL	• "Brinda soluciones seguras en el almacenamiento lo cual impide una perdida en la memoria." (Adolfo, 2012) • Contiene contraseña cifrada la cual brinda protección a los datos que se almacenan. • Velocidad al realizar las operaciones, lo que hace que sea un gestor (SGBD) con mayor rendimiento. • Facilidad su configuración e instalación ya que soporta una gran variedad de sistemas operativos.	• MySQL no puede trabajar de manera eficiente con bases de datos a gran escala. • En MySQL el mecanismo de almacenamiento de datos (MyISAM) no es compatible con las transacciones y es propenso a la corrupción de datos. (Tecnología, 2016) • En el sitio oficial de MySql la documentación no detalla el uso de todas las funcionalidades; se debe recurrir a información de sitios como blogs, foros o sitios no confiables.

Cassandra	• Optimiza las aplicaciones online de los negocios. • Corrige los errores que se presentan y mantiene una operatividad constante. • Contiene datos flexibles e Impide la perdida de información. • Maneja un lenguaje sencillo similar al de SQL. • La instalación típica de Cassandra es realmente sencilla, solo se requiere descargar desde la página oficial la carpeta que contiene todos los archivos de configuración y ejecución de la base de datos, pegarla en la raíz del disco C:\\ e iniciar el servicio con el comando “Cassandra” por medio de la línea de comandos de Windows. • Cassandra permite mayor agilidad y menos consumo de recursos de la máquina permitiendo la interacción por la línea de comandos de Windows, dado que no se utiliza una interfaz gráfica. • Cassandra utiliza el lenguaje de consultas CQL el cual es similar a SQL pero más limitado, es decir, ofrece menos opciones para la manipulación e interacción con los datos, y permite que los usuarios familiarizados con bases de datos como Sql Server o Mysql tengan el contexto básico del lenguaje a utilizar.	• Con el fin de apoyar las características de fiabilidad y coherencia, los desarrolladores deben implementar su propio código, lo que agrega más complejidad al sistema. • Cassandra por ser NoSQL no admite funciones SQL haciendo como los son los JOINS. • Como Cassandra es relativamente nueva en el mercado cuenta con muy poca documentación y esto hace que, en la implementación, se requiera invertir más tiempo para la comprensión de sus funcionalidades.

3.3.2 Tabla Comparativa métodos de seguridad en base de datos

Tabla 10. Tabla Comparativa métodos seguridad BD.

Método/ Motor	**SQL Server**	**MySQL**	**Cassandra**
Control de Acceso	• Utiliza seguridad basada en roles que facilita la administración de autorizaciones, la	• En el motor MySQL su nombre cambia y se denomina Sistema de privilegios de acceso.	• Cassandra Utiliza seguridad basada en roles, por medio de funciones de base de datos, que pueden

	separación de responsabilidades y la asignación de menor cantidad de privilegios. • Los permisos de los roles fijos no se pueden modificar. • Los roles definidos por el usuario permiten que se asignen permisos más específicos de acuerdo con las funciones que van a realizar los usuarios en el servidor y/o base de datos. • En la definición de los roles se establece que elementos se van a proteger, SQL Server permite asegurar las entidades que solicitan recursos para realizar acciones en el servidor y/o base de datos. • Para la creación de roles definidos por el usuario se genera dificultad al definir cuáles van a ser los roles y sus alcances.	• La información de los privilegios se almacena en las tablas de permisos user, db, host, tables_priv, columns_priv y procs_priv de la base de datos MySQL a la cual solo debe tener acceso el administrador de la base de datos. • MySQL valida el nombre de usuario y el equipo desde el que se conecta un usuario, esto permite tener varios usuarios con el mismo nombre asociados a diferentes equipos, cada uno con un conjunto de privilegios diferentes. • Si se otorgan privilegios CREATE y DROP para la base de datos *mysql* a un usuario, este puede eliminar la base de datos o las tablas en las que se almacenan los privilegios de acceso, generando daños en el sistema de privilegios ya sea permitiendo el acceso a usuarios no autorizados, asignando más privilegios de los que debe tener un usuario, entre otros.	representar un solo usuario o un grupo de usuarios, tanto en la administración de autenticación como en la de permisos. • En Cassandra no se encuentra bloqueada la opción para eliminar una base de datos de configuración del sistema, como puede ser "system_auth", que es la que almacena los roles y privilegios de acceso, un usuario por error podría realizar dicha acción y causar daños, eliminando la configuración de seguridad ya establecida.
Cifrado de datos	• El cifrado de datos se realiza mediante una infraestructura de cifrado jerárquico, se cifra por capas, cada capa cifra la capa anterior utilizando combinaciones de certificados, claves asimétricas y claves simétricas. Esta jerarquía hace que sea más difícil que un intruso pueda descifrar los datos. • SQL Server permite combinar diferentes	• SHA o SHA1 es un método más confiable que el MD5 ya que efectúa una suma de comprobación de 160 bits y MD5 realiza el proceso con 128 bits. • AES es un método considerado como seguro ya que se requiere de una contraseña para encriptar y desencriptar, lo que lo hace menos vulnerable. • El algoritmo AES es uno de los más utilizados y seguros actualmente en el mercado,	• El cifrado nodo a nodo con SSL protege los datos transferidos entre los nodos de un clúster, incluidas las comunicaciones de replicadas, utilizando SSL (Capa de sockets seguro) • El cifrado cliente nodo protege los datos en vuelo de las máquinas cliente a un clúster de base de datos que usa SSL (Capa de sockets seguros). Establece un canal seguro

	mecanismos y algoritmos de cifrado para realizar cifrado simétrico y asimétrico. • EL cifrado de datos es un método que consume muchos recursos, es necesario analizar qué datos realmente se deben cifrar y que tipo de cifrado utilizar, entre más seguro es el cifrado, más recursos se consumen. • El cifrado simétrico es más rápido que el cifrado asimétrico, sin embargo, tiene un nivel de seguridad más bajo. • El cifrado es más seguro si se incluye mayor longitud de clave y varias combinaciones de cifrado.	es usado por la agencia de seguridad nacional (NSA) de los estados unidos. (IBM, 2017)	entre el cliente y el nodo coordinador. • La interacción con el motor de base de datos de Cassandra se realiza a través de la línea de comandos de Windows, lo que permite mayor agilidad y menos consumo de recursos de la máquina, dado que no se utiliza una interfaz gráfica.

Las ventajas y desventajas expuestas en las tablas anteriores se lograron identificar durante la fase de investigación y con el análisis de los resultados obtenidos en la ejecución de los ataques, por ello se obtuvo el criterio necesario para identificar cuáles son los elementos más importantes, que se deben tener en cuenta durante la implementación de métricas y políticas de seguridad en una base de datos.

3.4 Fase 4: Metodología

Con los resultados y recomendaciones se concluye que la metodología utilizada en este proyecto fue demostrativa con efectos cualitativos y de aplicación sobre los datos, sobre las demostraciones y ataques realizados a las tablas de la base de datos utilizada, no basta con desarrollar una sola técnica o programa para realizar un aseguramiento de la información en los motores de bases de

datos; por ejemplo para el motor de bases de datos de SQL Server existen muchas más herramientas para realizar ataques de integridad de datos, sin embargo hay todo un mundo de oportunidades para los motores de bases de datos que no son licenciados o llamados de uso gratuito; de acuerdo a los resultados encontrados en el motor de bases de datos Cassandra es el que posee menos documentación y necesita un alto grado de conocimientos técnicos para configurar e implementar las medidas de administración necesarias para prevenir ataques y llegar a los resultados que se esperan.

Es importante entender que en un mundo globalizado las empresas deben poner mayor atención a la seguridad en sus bases de datos para que cada día identifiquen más vulnerabilidades para que así la información no esté en manos equivocadas; con el desarrollo de este proyecto se da un primer paso para reducir las brechas de seguridad en la información e identificar las buenas prácticas para el uso de la metodología con resultados eficientes.

Debido a la experimentación y lo mencionado anteriormente se propone el siguiente conjunto de procesos de metodología para implementar en cualquier organización que use o contenga información en bases de datos con el fin de aumentar la protección de esta y la reducción de vulnerabilidades de la empresa.

Propuesta y validación de metodología para asegurar la información en una base de datos

1. Análisis de roles, perfiles y usuarios

En esta fase de la metodología se pretende identificar los roles necesarios para la organización de acuerdo con cargos desempeñados en las áreas, con el fin de asignarles perfiles de niveles de seguridad.

1.1 Crear matriz de roles

En la organización se debe tener el registro de los roles dividido y categorizado de acuerdo a las áreas de la organización dentro de una matriz en la cual se pueda identificar, el nombre del rol, el área, el perfil requerido y el nivel de seguridad de este.

1.2 Establecer permisos de seguridad para los Roles (CRUD)

Establecer los permisos requeridos a cada rol, identificando el área y las funciones que se van a desarrollar, de acuerdo a estas necesidades se asignan dichos permisos (lectura, escritura, edición y eliminación)

1.3 Categorizar roles, perfiles y usuarios

Definir permisos, roles y usuarios, para poder clasificarlos en función a sus responsabilidad y actividades requeridas por la organización, esto permite delimitar el nivel de privilegios (Lectura, escritura, edición y eliminación) que se otorgaran.

2. Establecer políticas, procesos y métricas de seguridad

Lograr un posicionamiento de seguridad de la organización en base a normas y procedimientos, fomentar el desarrollo de procesos y técnicas que logren un mayor grado de fiabilidad a los procesos que con llevan la manipulación de información y uso de los datos, permitiendo la métrica de seguridad a los procesos internos.

2.1 Analizar vulnerabilidades de la organización.

Identificar los activos más susceptibles a los riesgos, como suelen ser, clave asimétrica, clave simétrica, contraseñas, rol de aplicación, usuario; e identificar posibles amenazas como la perdida de información o corrupción de datos.

2.2 Recopilar necesidades de seguridad (comité de seguridad).

En esta etapa es necesario contar con un comité de seguridad en cual se llevaran todas las decisiones a tomar, por ende si la organización no cuenta con este debe evaluarse su creación,

Por medio de las métricas de seguridad, reporte de logs entre otros se pueden identificar cuáles son las necesidades que se requieren suplir y llevar al comité de seguridad para el aseguramiento de la información, ya sea que estas se conviertan en proyectos de aseguramiento ante la interrupción, intercepción, modificación de la información custodiada.

2.3 Analizar amenazas a la información.

Identificar la probabilidad de que una amenaza se materialice por medio de canales de información o interacción con la base de datos, para reducir la probabilidad de las amenazas, se debe evitar errores como es el uso excesivo de privilegios innecesario, información sensible sin custodiar y falta de auditoría en los procesos de seguridad.

2.3.1 realizar análisis cualitativo y cuantitativo de las amenazas.

Identificar las amenazas que pueden vulnerar la seguridad e integridad de los procesos, reuniendo información sobre la probabilidad que la amenaza se materialice; estableciendo un valor específico de cómo y cuánto, puede afectar dichos procesos y el impacto que podría ocasionar.

2.4 Analizar riesgos de la organización que impacten la información o al sistema gestor de bases de datos

Se debe realizar un análisis de las vulnerabilidades plenamente identificadas dentro de la organización, clasificar el tipo de riesgo (Alto, Medio, Bajo) y los controles de seguridad informática que se tienen establecidos, para lograr presentar mejoras en los procesos y obtener una correcta administración de los riesgos con un plan de acción y seguimiento frecuente.

2.5 Crear políticas de seguridad

Se deben crear medidas que ayuden a mantener la confidencialidad de los datos sensibles de una organización, empresa o individuo, establecer parámetros y normas de cumplimiento dentro de los procesos.

2.5.1 Crear política de cifrado de información

- Identificar la información crítica que se deba proteger en la organización.
- Definir el tipo de cifrado que se va a utilizar.
- Generar una técnica de Cifrado.
- Validar la infraestructura con la que se cuenta y definir se está permite el uso de sus recursos sin perder rapidez.
- Establecer que usuarios van a tener acceso a la información

2.5.2 Crear política de creación de contraseña

Establecer medidas de seguridad para crear una contraseña, utilizando una combinación de mayúsculas, minúsculas, caracteres especiales, alfanuméricos, establecer una la longitud de

caracteres permitidos, no permitir palabras comunes ni información personal, no repetir la contraseña y establecer un tiempo de caducidad.

2.5.3 Crear política de cambio de contraseña

Determinar el tiempo de caducidad para realizar el cambio de la contraseña, establecer un límite de contraseñas repetidas y denegar el acceso a usuarios que no cambien su contraseña dentro del periodo permitido.

2.6 Realizar campañas de cultura y procesos de seguridad

Promover la cultura en la seguridad de la información dentro de la organización, teniendo en cuenta componentes como: concientización, entrenamiento y educación, dar a conocer los temas de importancia a las personas dentro de la organización, divulgar los principales riesgos de la seguridad de la información y fomentar el uso de las buenas prácticas de seguridad en la información, realizar ejercicios de reinducción con el fin de mantener el fulo de conocimientos en la organización.

2.6.1 Campaña y proceso de control de acceso

Informa a las personas dentro de la organización, que cada usuario cuenta con credenciales de accesos a los diferentes sistemas que se maneja, promoviendo que por seguridad las credenciales de acceso no deben ser compartidas con terceros.

2.6.2 Campañas de aprobación de peticiones de control de acceso

Divulgar que las peticiones de control de accesos tienen ciertos lineamientos, esta debe ser solicitada por personal autorizado, contar con los formatos de apoyo para dicha solicitud, requiere de la aprobación de un analista de control de accesos.

2.6 Crear métricas de seguridad

Establecer los límites permisibles de la seguridad en la información además de la administración de riesgos con planes de acción y cumplimiento alineado con los objetivos organizacionales.

2.6.1 Nivel de seguridad de la contraseña

Verificar si se está cumpliendo con la política de creación de contraseñas.

Validar que el proceso de autenticación si sea efectivo al momento de ser ejecutado por los usuarios e informar el nivel de seguridad de la contraseña al usuario y al administrador de usuarios.

2.6.2 Nivel de seguridad de la información (baja, media, critica, muy crítica)

Identificar la información de la organización y categorizarla de acuerdo a su nivel de importancia dentro de la organización para ello se debe auditar si la información crítica y sensible de la organización se encuentra accesible de acuerdo a las políticas de seguridad establecidas; identificando el nivel de riesgo al que se encuentra expuesto.

2.6.3 Conexiones simultáneas y conexiones consecutivas

Se debe generar un registro y trazabilidad de conexiones de usuario, con el fin de analizar la ubicación de estas conexiones en un corto periodo de tiempo, además de las múltiples conexiones que se deben permitir en la organización por un usuario crítico.

2.6.4 Alertas de seguridad

Definir alertas de seguridad que notifiquen el acceso de un usuario no autorizado al sistema, modificación de datos custodiados, alteración de privilegios de roles, eliminación de información, denegación de servicios, múltiples intentos fallidos de autenticación desde un mismo origen, accesos a honey Pots.

3. Mecanismos de cifrado de la información

En esta fase de la metodología se debe tener en cuenta para la organización aspectos como, el volumen de información sensible o delicada que se debe cifrar, la forma en que los usuarios van a acceder a dicha información, la capacidad de la plataforma tecnológica y el rendimiento mínimo que se debe garantizar; conforme a esto definir qué tipo de cifrado utilizar: simétrico, asimétrico o una combinación de ambos y los mecanismos más adecuados para el tipo de cifrado seleccionado.

3.1 Establecer herramientas de cifrado de la información.

Definir que herramientas se van a utilizar para el proceso de cifrado de datos, mediante un análisis de alternativas que indique la viabilidad entre adquirir una herramienta disponible en el mercado y desarrollar una propia acorde a las necesidades específicas de la organización. (Esto debe realizarse por el comité de seguridad)

3.2 Establecer mecanismos de cifrado de la información.

De acuerdo con las herramientas seleccionadas para el proceso de cifrado se definen los mecanismos que se van a utilizar.

3.3 Custodia de la información

Determinar si existe información confidencial que deba ser custodiada y establecer un proceso que permita proteger dicha información, mediante la creación de restricciones que garanticen que la información solo sea consultada por los usuarios autorizados, esta debe ser almacenada utilizando un mecanismo de cifrado adecuado y una ruta confidencial.

3.4 Flujo de aprobación de cifrado y descifrado de la información.

Establecer el flujo para el proceso de aprobación de cifrado y descifrado de la información,

Mediante la inclusión del personal afectado, la justificación de cifrado o descifrado, los parámetros bajos los cuales se realiza la acción, el tipo de cifrado que se debe realizar y si la información debe ser custodiada.

4. Pruebas de vulnerabilidad

En esta fase se pretende auditar la seguridad del sistema por medio de la ejecución de pruebas de vulnerabilidad de caja gris, caja negra, caja blanca, honey pots y pruebas de humo.

Es necesario determinar la periodicidad con la que se van a ejecutar dichas pruebas y quien las va a ejecutar, un especialista interno o externo a la organización.

4.1 Análisis de los resultados de las pruebas.

Analizar los resultados obtenidos durante la ejecución de las pruebas y definir planes de acción que permitan mitigar las vulnerabilidades encontradas.

5. Integración con la base de datos

Establecer el proceso de comunicación y petición de información a la base de datos con el fin de que cualquier aplicación, sistema o desarrollo obtenga información a través de los canales habilitados por mi base de datos.

5.1 Análisis de canales de conexión permitidos

Restringir los canales de comunicación permitiendo solo el intercambio de información con los que se necesiten para el correcto funcionamiento de todos los procesos en la organización.

5.2 Solicitudes de petición de la información

Se deben establecer medidas y privilegios a los canales de acuerdo a la necesidad por la cual fue creada, con el fin de entregar la información permitida por dicho canal y establecer el sentido en que debe ir la información.

5.3 Confiabilidad

Asegurar el cumplimiento de las políticas y métricas establecidas en las fases anteriores

6. Creación de usuarios

Definir un proceso mediante el cual se creen los usuarios de forma segura, que permita validar la autenticación, la autorización y los privilegios.

6.1 Crear y editar usuarios

Definir validaciones para garantizar la autenticidad de un usuario nuevo o existente.

Contar con la aprobación de creación o modificación del usuario.

Identificar el perfil y el rol al cual va a pertenecer.

Establecer validación de actividad del usuario (vacaciones, licencias, entre otros) y caducidad.

6.2 Validación de usuarios creados

Se deben establecer procesos y acciones periódicas por las cuales se identifiquen los cambios en los usuarios ya sea por periodos de inactividad conocidos o eliminación permanente de la cuenta.

6.3 Flujo de aprobación de creación de usuarios

Establecer el flujo para el proceso de aprobación de creación de usuarios, el cual debe incluir la los usuarios encargados de aprobar o rechazar, los parámetros bajos los cuales se va a validar la autenticación, el rol, el perfil y la autorización de los privilegios de dicha cuenta.

7. Auditoria

Realizar autoevaluaciones y seguimiento a los controles y políticas establecidas, automatizar procesos que permitan identificar fácilmente los cambios no autorizados;

Generar trazabilidad a actividades que afecten la integridad de datos sensibles.

7.1 Procesos de seguimiento y control (trazabilidad de la información)

Realizar seguimiento y trazabilidad a las solicitudes de petición de la información de datos sensibles, generar reportes de las peticiones de los usuarios que interactuaron dicha información con la finalidad de sacar estadísticas y métricas de posibles vulnerabilidades, generar datos estadísticos para el análisis y toma de decisiones.

7.2 Validación de cumplimiento de las políticas de seguridad

Realizar auditoria de cada una de las políticas de seguridad que se tienen establecidas en la organización, verificar que las campañas del proceso de control de acceso y aprobación de peticiones de acceso se realicen periódicamente en cada una de las áreas de la organización y validar que la cultura de seguridad de la información se implemente continuamente.

7.3 Validación de registro de casos críticos, amenazas y alarmas de la base de datos

Establecer un procedimiento formal para el registro de los eventos presentados, el tipo de evento, su contención o su impacto, tiempo de ejecución daños generados e información afectada.

7.4 Análisis de los logs de información

- Establecer una técnica de las actividades que conllevan a la generación del análisis de la información obtenida en los logs, para este caso se pueden analizar la leyes de protección de ciberseguridad y forense de datos del país donde se encuentra la organización.
- Se debe validar que los eventos y hallazgos encontrados registrados en los logs, de las operaciones que se realizan o interactúa con la base de datos de forma sospechosa, sean de útil interpretación y como objeto principal se establece monitorear los servicios asociados a la información crítica de la organización.
- Se debe establecer el proceso de respaldo de los logs que contengan alertas de acciones no autorizadas para mantener una trazabilidad del proceso de seguridad.

7.5 Validación de usuarios creados versus empleados activos de la organización

Auditar los registros de creación de usuarios y realizar un comparativo de la fecha de salida de los usuarios retirados contra los usuarios activos con el fin de identificar que usuarios retirados se

encuentran activos y la justificación del porque no han sido desactivados, ya que estos generan un alto riesgo de intrusión a la base de datos.

7.6 Validación y actualización de perfiles, permisos y roles.

Auditar los registros de actualización de cambios de cargo o movimientos entre áreas de los usuarios y validar que la información de roles y permisos en el sistema sea consecuente, para evitar riesgos de seguridad al filtrar información a áreas y personal no autorizado.

Validar que los usuarios retirados no pertenezcan a ningún rol activo en el sistema.

Validar que la información de roles permisos y usuarios este actualizada y corresponda con los usuarios activos y las funciones que desempeña actualmente.

Validar que los cambios realizados a los roles, perfiles y permisos de acceso haya contado con el flujo de aprobación debido.

8. Mejora continua del proceso de seguridad

Utilizar los datos estadísticos generados de los registros y trazabilidad de los casos presentados y analizados con el fin de fortalecer y reevaluar la efectividad de las políticas, métricas y procesos de seguridad de la compañía.

8.1 Actualización a las políticas de seguridad

Para este ítem se deben tener reuniones frecuentes del comité de seguridad donde se presenten los casos y evidencias de las amenazas, intrusiones, falsos positivos y demás hallazgos que interactúen con el motor de base de datos y generar nuevas políticas, actualización de las actuales y eliminación de las obsoletas.

8.2 Actualizar los procesos de acuerdo a los hallazgos y vulnerabilidades identificados en el análisis de los logs de información

Se deben realizar reuniones de comité de seguridad con el fin de reevaluar la efectividad de los procesos implementados y los casos presentados para generar actualizaciones a los mismos y reducir la brecha de vulnerabilidad del motor de bases de datos.

8.3 Actualización a las políticas de cifrado

Mantener actualizadas las políticas de cifrado establecidas, realizando los cambios necesarios para estar al día con los avances tecnológicos y nuevos desarrollos para la ejecución de ataques.

8.4 Capacitación de seguridad a usuarios.

Definir planes de sensibilización, temas a tratar para que los usuarios identifiquen los riesgos y amenazas que afectan la confidencialidad, integridad y disponibilidad de la información.

Establecer una metodología de capacitación perióca que permita evaluar, medir y cuantificar la participación de los asistentes.

Fomentar la capacitación continua de los usuarios para fortalecer el cumplimiento de las políticas de seguridad establecidas.

4. Resultados y Discusión

- Se identificó por medio de la ejecución de las fases de investigación y experimentación, que en el motor de base de datos SQL Server, es posible elevar el nivel de seguridad al establecer un proceso de seguridad más complejo, en comparación con los otros motores seleccionados, con ayuda de la implementación de la jerarquía de cifrado, que permite cifrar por capas y utilizar diferentes combinaciones de certificados, claves asimétricas y claves simétricas, lo que conlleva a la creación de un proceso personalizado que no se puede identificar fácilmente por un agente externo, proporcionando mayor integridad y confidencialidad a la información almacenada.

- Dentro de la evidencias y pruebas analizadas se pudo considerar que establecer políticas de creación de claves seguras reducen las vulnerabilidades frente a los ataques de control de acceso, lo anterior es justificado ya que el proceso que se debe realizar para descifrar una clave que contiene una longitud de cadena amplia, requiere de un esfuerzo mayor en tiempo y recursos, lo cual en ocasiones podría generar que el intruso pierda el interés en realizar el ataque o simplemente no lo logre.

- Utilizando el mismo tipo de ataque de control de acceso, se identificó que, al utilizar Kali Linux se obtuvieron mejores resultados, debido a que este es un sistema operativo que cuenta con una recopilación de las mejores herramientas diseñadas para la auditoria y seguridad informática en general, lo que hizo la experiencia de intrusión más eficiente, pero es de aclarar que esto requiere de conocimientos básicos en las herramientas y sus funcionalidades.

- En alineación con la práctica anterior haciendo uso de las herramientas Johnny y HexorBase, se logró desencriptar los hashes de las contraseñas de los usuarios creados de manera más eficiente y efectiva en los tres motores de bases de datos bajo configuraciones básicas y estándar de los motores de bases de datos.

- Con base en la ejecución de los ataques realizados en los tres motores seleccionados, se identificó que un ataque de fuerza bruta que utiliza diccionario o lista de palabras es muy efectivo, cuando se tienen registradas claves con un nivel de seguridad bajo para el ingreso a la base de datos, esto facilita a un intruso el acceso, puesto que en la web es posible encontrar diversos diccionarios, frecuentemente actualizados, algunos muy completos con un amplio listado de posibles contraseñas y lo más notorio es su facilidad para adquirirlos.

- El ataque de control de acceso en mysql con la herramienta HexorBase, demuestra que la seguridad implementada en este motor de base de datos es muy efectiva para evitar el acceso de un usuario no autorizado, ya que para acceder al servidor es necesario contar con el nombre o dirección IP del equipo además del usuario y la contraseña, sin estos tres datos no es posible la conexión, a diferencia de lo evidenciado con los motores SQL Server y Cassandra que solo requieren de usuario y contraseña.

- De acuerdo con los resultados obtenidos durante el proceso de ataques en SQL Server, se evidencia que el cifrado de datos implementado es muy efectivo para proteger los datos almacenados, puesto que para llegar a los datos cifrados es necesario descifrar cada una de las capas creadas y en la web no se encuentra una herramienta que realice por completo el mismo proceso, el cual puede ser definido por el usuario y varía según su criterio, es decir, el cifrado

de datos se puede realizar de diferentes formas y sería necesario comprender como fue realizado para lograr descifrar los datos.

- Cassandra entre las bases de datos NoSQL es la más popular y distribuida, debido a que proporciona alta estabilidad y disponibilidad, sin embargo, como es relativamente nueva en el mercado cuenta con muy poca documentación, esto hace que el proceso planteado para la implementación de seguridad en base de datos, requiera de mayor análisis y practica; Adicionalmente durante la investigación se identificó que solo se pueden configurar parámetros básicos de seguridad y que es necesario el desarrollo de nuevas funcionalidades, por ejemplo, funciones de cifrado de datos, que no existen dentro de este sistema gestor de bases de datos.

5. Conclusiones, Recomendaciones y Trabajo Futuro

5.1 Conclusiones

- Durante el proceso de investigación se identificó que no es posible asegurar por completo la información de una base de datos, ya que los riesgos no se pueden eliminar en su totalidad, se recomienda administrar el riesgo de tal forma que las vulnerabilidades disminuyan con la ejecución de métodos de seguridad; en la investigación y desarrollo del proyecto, se recopilo la información necesaria para llevar a cabo el proceso de experimentación de los métodos de seguridad en los tres motores de base de datos seleccionados, a partir de esto se obtuvo una serie de resultados que fueron analizados con el fin de identificar la estructura de los procesos que se deben tener en cuenta para incrementar el nivel de seguridad de la información en una base de datos; se logró crear una propuesta de metodología que incluye una estructura de procesos que se deben ejecutar para reducir las brechas existentes de seguridad, de acuerdo a la experiencia adquirida se recomienda seguir los pasos definidos en la propuesta de la metodología ya que esta fue creada con base en la experimentación realizada de los métodos de control de acceso y cifrado de datos, acorde con el análisis de los resultados formulados estos métodos proporcionan mayor seguridad a la información, así mismo se buscó que los procedimientos propuestos se puedan aplicar a cualquier organización que intente mejorar la seguridad de la información almacenada en una base de datos.

- Mediante el proceso de ataques realizado en la experimentación, se logra identificar que la intrusión a una base de datos se puede lograr, con la ejecución de ataques de diccionario de datos, que consisten básicamente en un método de prueba y error para validar con cuál de las

claves registradas en el diccionario se logra acceder, y ataques de fuerza bruta y cracking, que además del método de prueba y error utilizan un algoritmo de descifrado para lograr descifrar una clave encriptada.

- Las intrusiones a las bases de datos se pueden lograr mediante la ejecución de Ataques de diccionario de datos, que consisten básicamente en un método de prueba y error para validar con cuál de las claves registradas en el diccionario se logra acceder, y ataques de fuerza bruta y cracking, que además del método de prueba y error utilizan un algoritmo de descifrado para lograr descifrar una clave encriptada.

- A lo largo de la investigación se logró recopilar evidencias que identifican los factores que influyen en las buenas prácticas para asegurar la información en una base de datos; la seguridad depende de un conjunto de aspectos que al ir de la mano llevan al correcto funcionamiento de los métodos de seguridad implementados, que incluyen la verificación de la forma en que se instaló la base de datos, la definición de los roles y sus alcances.

- Definir políticas de creación de claves seguras, realizar campañas de sensibilización y procesos de seguridad, permiten establecer una cultura de seguridad de la información basada en la importancia de proteger los datos y mitigar las vulnerabilidades a las que puede estar expuesta una base de datos. De modo que al no establecer dichas políticas las vulnerabilidades del sistema incrementan dando paso a posibles ataques que se pueden lograr fácilmente utilizando las herramientas disponibles en la web, lo cual se comprobó en la práctica desarrollada. Cabe señalar que el uso de las herramientas requiere de una amplia investigación para lograr un buen manejo de estas y conseguir un ataque exitoso.

- En el ejercicio de la experimentación se logró identificar que los métodos de control de acceso y cifrado de datos suelen tener un grado de dificultad mayor de acuerdo al motor de base de datos en el que se esté implementando, ya que los procesos y las técnicas de seguridad son diferentes en cada uno, al comparar el proceso de cifrado en cada motor se puede decir que en SQL Server es más seguro ya que este permite realizar un cifrado más complejo por medio de la infraestructura de cifrado jerárquico dispuesto para ello, en MySQL el nivel de seguridad de los datos depende del método de cifrado que utilice la función seleccionada; mientras que en Cassandra se encuentra disponible el cifrado para proteger la comunicación entre nodos, sin embrago, no cuenta con funciones que permitan almacenar los datos cifrados y esto hace que la información quede expuesta ante el ataque de un intruso.

- Durante el proceso de investigación se logró identificar, que los métodos de seguridad que mayor impacto positivo producen en el aumento del nivel de seguridad son: el control de acceso y el cifrado de datos, dado que con estos se logra asegurar los procesos más vulnerables, como son el acceso al servidor y a los datos almacenados. Al implementar estos dos métodos en conjunto se reducen las probabilidades de que una amenaza se materialice.

5.2 Recomendaciones

- SQL Server a través de las opciones mapear certificado y mapear clave asimétrica permite configurar otro nivel de autorización en el acceso a un objeto asegurado (tabla, procedimiento almacenado, vista, entre otros), el cual se puede conceder temporalmente a un usuario o proceso de conexión y se revoca cuando ya no se necesite, este proceso permite mantener el

control de los permisos que ya están asignados, para evitar modificarlos cuando se requiere un permiso adicional de forma temporal a un usuario.

- De acuerdo con el desarrollo de la experimentación en la fase de ataques para el método de control, se recomienda utilizar las aplicaciones que contiene el sistema operativo Kali Linux, puesto que, con este se obtuvieron mejores resultados, teniendo en cuenta que para llevar a cabo la ejecución de ataques es necesario contar con algunos conocimientos básicos en las herramientas y sus funcionalidades, las aplicaciones que se emplearon dentro de la experimentación utilizan la técnica de ataque de fuerza bruta, ya que este es el más indicado para realizar ataques de control de acceso.

A continuación, se mencionan las herramientas utilizadas en el proyecto y que se recomiendan por su efectividad, de acuerdo con el resultado obtenido en la experimentación.

  - Crunch: funcionalmente es muy eficiente y posee una usabilidad que brinda comodidad y entendimiento para realizar el proceso de creación de diccionarios con tipos de claves específicas, que no son encontrados en la web y que se requieren para realizar un ataque. Sin embargo, se debe contar con una máquina de gran capacidad de procesamiento, para que la aplicación sea capaz de finalizar la creación del diccionario con todas las líneas de las posibles combinaciones, y permutaciones encontradas.
  - HexorBase: es muy eficiente porque permite auditar y realizar ataques de fuerza bruta a múltiples servidores de bases de datos (MySQL, SQLite, Microsoft SQL Server, Oracle y PostgreSQL) simultáneamente desde una ubicación centralizada. Para que el ataque sea exitoso es necesario contar con dos diccionarios que contengan el usuario y la clave del motor de base de datos al cual se quiere acceder.

- Ohnny: la interfaz gráfica de esta aplicación permite descifrar contraseñas sin necesidad de escribir líneas de código, lo que hace que la interacción con el usuario sea más comprensible. Esta también requiere de una máquina de gran capacidad de procesamiento para que el proceso culmine al 100%.
- Cain & Abel: se recomienda utilizar esta aplicación para el método de cifrado de datos en MySQL, ya que ofrece diferentes opciones de cracking empleando algoritmos de cifrado tales como: MD2, MD4, MD5, SHA1 y SHA2, los cuales son usados comúnmente en el cifrado de datos.

- Para el método de cifrado de datos en SQL Server se recomienda utilizar el script desarrollado, ya que de acuerdo con la investigación no se encuentra una herramienta que realice todo el proceso para descifrar los datos que fueron cifrados por capas (cifrado jerárquico), es decir, es necesario descifrar cada capa hasta llegar a los datos cifrados y con las aplicaciones disponibles solo sería posible descifrar una capa a la vez, esto requiere de un esfuerzo mayor en tiempo y recursos.

- Se recomienda no utilizar comodines al momento de asignar los privilegios de acceso en Mysql, ya que estos permiten otorgar privilegios en exceso a los usuarios y hacen que la seguridad del acceso al servidor se debilite, permitiendo la manipulación de la información a usuarios de forma indebida.

- Para la instalación de Cassandra se recomienda llevar a cabo las instrucciones dadas en la documentación de la página oficial de Cassandra, ya que brinda información confiable y precisa de las funcionalidades que están incluidas en la última versión disponible.

- Para implementar un cifrado seguro, se debe realizar un análisis donde se evalué, si se requiere mayor rendimiento o si se desea conseguir mayor protección en los datos, ya que la implementación de un cifrado seguro genera mayor protección a los datos, y a su vez mayor consumo de recursos y pérdida de rendimiento.

5.2.1 Dificultades durante el desarrollo de la fase de experimentación se presentaron contratiempos en el momento de realizar la instalación y el uso de algunas herramientas; se nombran a continuación con el propósito de que se tengan en cuenta en la implementación de futuros proyectos.

5.2.1.1 Instalación VirtualBox este software de virtualización se utilizó para instalar el sistema operativo Kali Linux, ya que se disponía de equipos con sistema operativo Windows.
El proceso de instalación de VirtualBox es sencillo, una vez descargado el instalador de la página oficial se siguen los pasos del asistente y al finalizar la aplicación se inicializa de forma correcta, hasta que se ejecuta y se pretende instalar una imagen Kali Linux, es en este punto donde se presenta el error que se visualiza en la *Ilustración 46.*

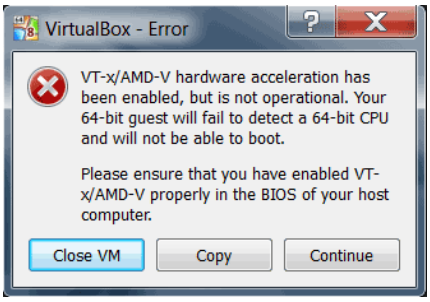

*Ilustración 47.*Error VirtualBox.

Para solucionarlo basta con habilitar la virtualizar en la opción de seguridad de la BIOS del computador, reiniciar y volver a inicializar Virtualbox.

En el administrador de tareas de Windows en la opción Rendimiento se puede observar que la virtualización se encuentra habilitada.

Nota: en algunos equipos Lenovo con referencia Z480 no permite el ingreso a la BIOS por un defecto de fábrica.

Los requerimientos mínimos que debe tener la maquina donde se va a instalar Kali Linux (utilizando VirtualBox) para evitar lentitud y posibles bloqueos son los siguientes:

Procesador: Core i7

Memoria Ram: 12 GB

Tipo de sistema: sistema operativo de 64 bits, procesador x64

Espacio en disco duro: 30 GB

5.2.1.2 Aplicación Crunch de Kali Linux durante la realización de las pruebas con la aplicación Crunch se intentó generar un diccionario con una gran cantidad de líneas, lo que genero un bloqueo en la máquina virtual que impedía que se ejecutaran correctamente otras acciones, inicialmente se intentó cerrar la aplicación para finalizar el proceso y continuar, sin embargo, solo se logró bloquear aún más la maquina hasta el punto de tener que reiniciarla, en este caso al iniciar de nuevo la maquina generaba error y no fue posible volver a iniciar Kali Linux, por lo tanto fue necesario realizar una nueva instalación.

Para evitar realizar una nueva instalación de Kali Linux, se recomienda matar el proceso que este ejecutando Crunch, por medio del Shell de comandos, se debe consultar el PID del programa y matar ese proceso, los comandos son los siguientes:

ps -ef | grep: Crunch se consulta el número del proceso que ejecuta crunch

Kill -9 1985: se mata el número del proceso que corresponde a crunch

5.2.1.3 Instalación Cain & Abel la instalación de la herramienta Cain & Abel requiere que se desactive el antivirus y el firewall de Windows Defender, ya que el tenerlos activos genera el mensaje de error: “Error de sistemas, no se encontró packet.dll” lo que impide finalizar la instalación.

5.2.1.4 Archivos de configuración de Cassandra cuando se realiza cualquier modificación en el archivo de configuración cassandra.yaml es necesario reiniciar la maquina por completo para que cassandra tome los cambios, no basta con reiniciar los servicios de cassandra.

Habilitar la autenticación y autorización de Cassandra

El proceso de creación de roles requiere que la autenticación y autorización se encuentre habilitado, para ello la página oficial de Cassandra recomienda que antes de modificar el archivo cassandra.yaml, se cambie el factor de replicación del system_auth, que por defecto trae el valor 1, por un factor de replicación de 3 a 5 por Data center o nodo. Se debe tener en cuenta que este cambio solo se debe ejecutar cuando se tiene configurado más de un nodo, de lo contrario no es necesario.

5.3 Trabajo futuro

La elaboración del proyecto de investigación permitió identificar que los métodos de seguridad que generan mayor protección a la información son el control de acceso y el cifrado de datos, por ende lo más recomendable es realizar una implementación que contenga ambos métodos, sin embargo, esta implementación no es posible realizarla por completo en Cassandra, ya que este motor de base de datos no cuenta con una función que permita cifrar datos, como si lo hacen los motores SQL Server y MySQL, es por ello que como trabajo futuro se sugiere realizar una

investigación para llevar a cabo el desarrollo de nuevas funcionalidades que permitan implementar un cifrado de datos en el motor de base de datos Cassandra.

Referencias

ACISSI. (2015). Seguridad informática - Hacking Ético: Conocer el ataque para una mejor defensa. En ACISSI, *Seguridad informática - Hacking Ético: Conocer el ataque para una mejor defensa* (págs. 38-39). Barcelona: Ediciones ENI.

Andrades, J. A. (2012). *SlideShare*. Obtenido de https://es.slideshare.net/jabenitez88/8realizacion-de-pruebas-14981770

Apache Software Foundation. (2016). *Documentación Cassandra*. Obtenido de Documentación Cassandra: http://cassandra.apache.org/doc/latest/operating/security.html#roles

Avilés, G. G. (2015). Seguridad en Bases Datos y Aplicaciones Web. En G. G. Avilés, *Seguridad en Bases Datos y Aplicaciones Web* (págs. 21-25-28). No Definida: IT Campus Academy.

bofh28. (16 de 02 de 2014). *Crunch | Herramientas de prueba de penetración*. Obtenido de Crunch | Herramientas de prueba de penetración: https://tools.kali.org/password-attacks/crunch

Cassandra. (abril de 2018). *Casandra*. Obtenido de http://cassandra.apache.org/ y https://www.datastax.com/

Chávez, J. D. (12 de Julio de 2015). Principios Básicos de Seguridad en Bases de Datos . *Principios Básicos de Seguridad en Bases de Datos* . Maracay, Aragua, Venezuela: Universidad Politécnica Territorial del Estado Aragua.

douglaslMS. (30 de 03 de 2017). *Funciones de Servidor y base de datos en SQL Server | Microsoft docs*. Obtenido de Funciones de Servidor y base de datos en SQL Server |

Microsoft docs: https://docs.microsoft.com/en-us/dotnet/framework/data/adonet/sql/server-and-database-roles-in-sql-server

Ekiko, S. E. (14 de 02 de 2014). *HexorBase | Herramientas de prueba de penetración*. Obtenido de HexorBase | Herramientas de prueba de penetración: https://tools.kali.org/vulnerability-analysis/hexorbase

González, E. S. (2015). *Salvaguarda y seguridad de los datos. IFCT0310.* España: IC Editorial.

Gutierrez del Moral, L. (2013). Curso de Ciberseguridad y Hacking Ético. En L. G. Moral, *Curso de Ciberseguridad y Hacking Ético* (págs. 530-531-533-534). Sevilla: Punto Rojo Libros.

Isec auditors. (2017). *Test de Intrusión | Internet Security Auditors*. Obtenido de Test de Intrusión | Internet Security Auditors: https://www.isecauditors.com/test-de-intrusion

Linux, K. (abril de 2018). *Our Most Advanced Penetration Testing Distribution, Ever.* Obtenido de https://www.kali.org/

Macauley, E. (14 de 03 de 2017). *Jerarquía de cifrado | Microsoft Docs*. Obtenido de Jerarquía de cifrado | Microsoft Docs: https://docs.microsoft.com/es-es/sql/relational-databases/security/encryption/encryption-hierarchy

Microsoft. (2016). *MSDN*. Obtenido de MSDN: https://msdn.microsoft.com/es-es/eses/library/ms161953.aspx

Microsoft. (30 de 03 de 2017). *docs.microsoft*. Obtenido de docs.microsoft: https://docs.microsoft.com/en-us/dotnet/framework/data/adonet/sql/server-and-database-roles-in-sql-server

Microsoft. (27 de 11 de 2017). *La contraseña deben cumplir con los requisitos de complejidad*. Obtenido de La contraseña deben cumplir con los requisitos de complejidad:

https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-R2-and-2012/hh994562(v=ws.11)

Microsoft. (2017). *Microsoft*. Obtenido de Microsoft: https://technet.microsoft.com/es-es/library/cc753976(v=ws.11).aspx

Microsoft. (abril de 2018). *SQL Server*. Obtenido de https://docs.microsoft.com/es-es/, https://technet.microsoft.com/es-co y https://msdn.microsoft.com/es-es/

Microsoft. (2018). *TN Introducción al control de acceso*. Obtenido de TN Introducción al control de acceso: https://technet.microsoft.com/es-es/library/cc753976(v=ws.11).aspx

Montoro, M. (07 de 04 de 2014). *oxid.it Caín y Abel*. Obtenido de oxid.it Caín y Abel: http://www.oxid.it/cain.html

Moral, L. G. (2013). Definiciones y Terminología. En L. G. Moral, *Curso de Ciberseguridad y Hacking Ético 2013* (págs. 71-72-304). Sevilla: Punto Rojo Libros, S.L.

Moral, L. G. (2014). Curso de Ciberseguridad y Hacking Ético. En L. G. Moral, *Curso de Ciberseguridad y Hacking Ético* (págs. 530-531-533-534). Sevilla: Punto Rojo Libros.

Muñoz, A., & Hernandez, M. (2017). Análisis de la seguridad de las bases de datos orientadas a grafos. *Revista SIC: ciberseguridad, seguridad de la información y privacidad, ISSN 1136-0623, Vol. 26, Nº. 125 (Junio 2017), .*

MySQL. (abril de 2018). *MySQL 5.7 Reference Manual*. Obtenido de https://dev.mysql.com/doc/refman/5.7/en/

Oracle Corporation. (2010). *MySQL*. Obtenido de MySQL: http://ftp.tcrc.edu.tw/MySQL/doc/refman/5.0/es/privileges.html

Quezada, A. E. (28 de 06 de 2016). *Recopilar_Informacion_y_Realizar_un_Ataque_por_Fuerza_Bruta_contra_SQL_Server_utilizando_SQLPing_3*. Obtenido de Downloads - SQLSecurity Home: http://www.reydes.com/d/?q=Recopilar_Informacion_y_Realizar_un_Ataque_por_Fuerza_Bruta_contra_SQL_Server_utilizando_SQLPing_3

Sanchez, B. V. (2007). Seguridad en Bases de Datos.

Science Direct. (Abril de 2018). Obtenido de https://www-sciencedirect-com

Scopus. (abril de 2018). Obtenido de https://www.scopus.com/

Shinnok, A. C. (18 de 02 de 2014). *Johnny | Herramientas pruebas de penetración*. Obtenido de Johnny | Herramientas pruebas de penetración: https://tools.kali.org/password-attacks/johnny

Stallings, W. (2004). *Fundamentos de Seguridad en Redes. Aplicaciones y Estándares.* Madrid: Pearson Educación.

Torres, S., GONZALES, B., Adina, & Vavilova, I. (2013). La cita y Referencia Bibliografica: Guia basada en las normas APA 2012 . Buenos Aires, Argentina.

Web of Science. (abril de 2018). Obtenido de http://apps.webofknowledge.com

Yera, Á. C. (2007). Diseño y programación de bases de datos. En A. Cobo, *Diseño y programación de bases de datos* (págs. 7-8-9). España: Editorial Visión Libros.

FIRMA ESTUDIANTES Daniel González

Lisday Zapata d.

Yennifer Villa.

FIRMA ASESOR Gustavo Lacs

FECHA ENTREGA: 2018-05-03

FIRMA COMITÉ TRABAJO DE GRADO DE LA FACULTAD _____

RECHAZADO__ ACEPTADO____ ACEPTADO CON MODIFICACIONES_____

ACTA NO._____

FECHA ENTREGA: _____

FIRMA CONSEJO DE FACULTAD_____

ACTA NO._____

FECHA ENTREGA: _____